2004

Some studies on the multi-mesh architecture.

Nahid Afroz
University of Windsor

Some Studies on the Multi-Mesh Architecture

by
Nahid Afroz

A Thesis
Submitted to the Faculty of Graduate Studies and Research through the School of Computer Science in Partial Fulfillment of the Requirements for the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2004

Your file Votre référence
ISBN: 0-612-96116-8
Our file Notre référence
ISBN: 0-612-96116-8

Abstract

In this thesis, we have reported our investigations on interconnection network architectures based on the idea of a recently proposed multi-processor architecture, Multi-Mesh network. This includes the development of a new interconnection architecture, study of its topological properties and a proposal for implementing Multi-Mesh using optical technology.

We have presented a new network topology, called the 3D Multi-Mesh (3D MM) that is an extension of the Multi-Mesh architecture [DDS99]. This network consists of n^3 three-dimensional meshes (termed as 3D blocks), each having n^3 processors, interconnected in a suitable manner so that the resulting topology is 6-regular with n^6 processors and a diameter of only $3n$. We have shown that the connectivity of this network is 6. We have explored an algorithm for point-to-point communication on the 3D MM. It is expected that this architecture will enable more efficient algorithm mapping compared to existing architectures.

We have also proposed some implementation of the multi-mesh avoiding the electronic bottleneck due to long copper wires for communication between some processors. Our implementation considers a number of realistic scenarios based on hybrid (optical and electronic) communication. One unique feature of this investigation is our use of WDM wavelength routing and the protection scheme. We are not aware of any implementation of interconnection networks using these techniques.

Keywords: Multiprocessor Architecture, Interconnection Network, Network Parameters, Mesh Network, Multi-Mesh, 3D MM, Diameter, Connectivity, Routing, Optical Network, Optical Communication, WDM wavelength routed optical network, Optical Implementation of a network, Fault tolerant.

To my parents, brothers and sisters

Acknowledgements

I would like to take this opportunity to thank my thesis advisor Dr. Subir Bandyopadhyay for his direction, patience and cooperation throughout my graduate studies. I am grateful to him for the countless hours that he spent with me to help me understand the areas I studied during my research. Without his guidance, support, suggestions and pointers this work wouldn't be achieved. Special thanks to Dr. Bhabani P. Sinha who was a research scientist visiting our optical lab in summer 2003. He introduced me to the concept of interconnection network and provided all kinds of help whenever I needed. It is always a privilege to work with an excellent group of Dr. Subir Bandyopadhyay and Dr. Bhabani P. Sinha.

I wish to express my thanks to Dr. Alioune Ngom and Dr. Narayan Kar for their valuable time, cooperation and thoughtful suggestions that improved the quality of this thesis.

My special gratitude to my husband, Rabiul for his loving support and continuous encouragement. My little son Sindeed, I hope Sindeed will forgive me for the time I have spent on my studies when I was forced to leave him in day-care.

Last but not the least, I wish to thank and dedicate this thesis to my beloved parents, my brothers and sisters who offered their best wishes throughout my whole life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Multiprocessor Architecture

One of the prime objectives in designing computers has always been to build faster and more powerful machines. An obvious way to solve a problem faster is to use a network of a large number of processing units or computers, where the different processors solve a problem by working simultaneously on different parts of that problem [Ak89]. With the advances in technology, the cost and size of processors have been reduced tremendously so that it is now possible to use several thousands to millions of computers to build up a multi-processor system [Ak89]. The challenge on the hardwire side is to determine how these processors should be connected together for optimum performance.

1.2 Interconnection Network

A crucial part of designing a multi-computer architecture is to assure faster data communication to allow efficient sharing of data between the processors. Data communication is accomplished by sending messages through the computers by using shared memory or interconnection network [To94], [Ak89]. In this work, we will consider only interconnection networks.

The architecture of an interconnection network defines exactly which processors are connected to each other.

In an interconnection network, each processor has a memory and is interconnected to other processors with respect to a given topology. Figure 1.1 shows a general architecture of a multi-processor architecture where a number of processors and memory modules are connected by an interconnection network.

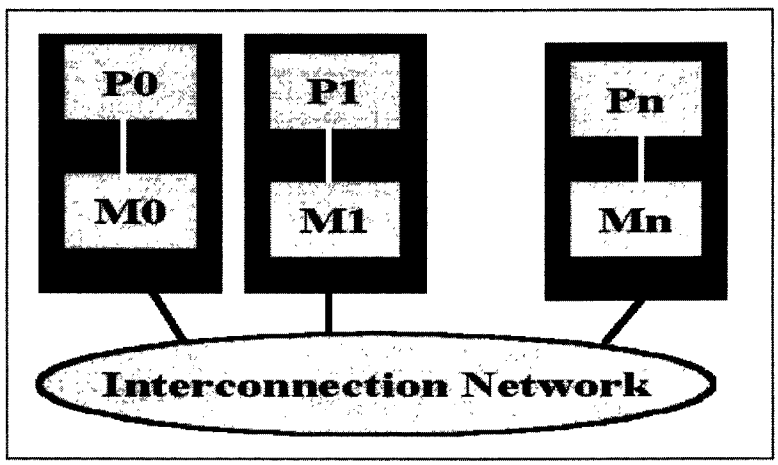

Figure 1.1: A multi-processor architecture

The architecture of the interconnection network in a multi-processor system has a crucial role in the performance of the multiprocessor system – both in terms of the speed of communication and the time to run an application.

Figure 1.2 shows an example of a simple multiprocessor architecture called tree, where a number of processors P1, P2,..., P7 are connected by interconnection network.

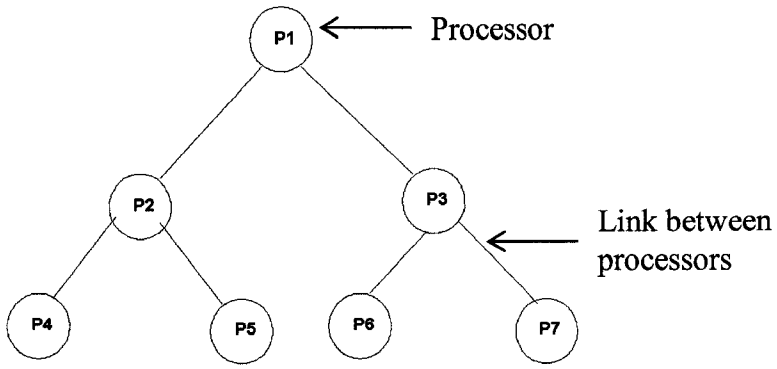

Figure 1.2: Tree interconnection network

In the last few decades, there has been a lot of effort in developing efficient multiprocessor interconnection architectures. Two-dimensional mesh [HwBr83], [St83], [Le92] is one of the most popular architectures due to its inherent simplicity and ease of algorithm mapping. Many variants of the two-dimensional mesh structure, e.g., torus, Illiac IV [HwBr83], [Le92], multi-dimensional mesh [Le92] have also been proposed in the literature in order to have a topology for more and more efficient computation in a parallel/distributed environment. Efficient mapping of many fundamental and most frequently used algorithms on variations of the mesh structure have been reported [DDS99], [DGS97], [HwBr83], [StCo91], [Le92].

In this thesis, we investigate the Multi-Mesh architecture [DDS99] - a recent proposal for multi-processor interconnection architecture that uses the 2-dimensional mesh as a basic building block. With the same number of processors and the same number of links as in the case of a torus, the Multi-Mesh (MM) topology has a much smaller diameter [DDS99] so that processors can communicate with each other quickly.

1.3 Optical Communication

Optical networks are used for data communication where signals carrying the data are in the form of light waves [ChKr93]. In an optical network, optical fibre is used as the media of transportation. Recently, there has been growing interest in developing optical networks to support the increasing bandwidth demands of multimedia applications, such as video conferencing and World Wide Web browsing [BBRM97]. According to Chamberlain and Krchnavek [ChKr93] optical networks have made significant contributions to the state of the art for long distance communications, including high reliability, low interference, security benefit and very high bandwidth. Traditionally, metal-based electrical connection has been used to realize interconnection networks. There are a number of limitations in this approach that we will review in chapter 2. For high-speed communication in interconnection networks, optical technology has been proposed as a better alternative to copper based communication [LoSu94a], [LoSu94b].

1.4 Work Reported in this Thesis

In this thesis, we report our investigations on mesh type interconnection networks based on the concept of the Multi-Mesh architecture. The main results are as follows:

1) We have proposed a new architecture that uses the 3-dimensional mesh as its building block rather than a 2-dimensional mesh as done in the Multi-Mesh [DDS99]. We have shown that our architecture has better topological properties compared to the Multi-Mesh architecture and that a number of algorithms can be efficiently mapped on the 3D MM network.

2) We have explored a number of possible approaches for implementing the Multi-Mesh architecture using opto-electronic technologies. There are two novel features of our approach:

    a. We have shown that WDM wavelength-routed networks may be used to realize some of the links.

    b. We have shown that single faults may be handled easily without increasing the number of optical paths used.

Our new topology which we call the 3D Multi-Mesh (3D MM) consists of n^3 three-dimensional meshes (termed as *3D blocks*), each having n^3 processors, interconnected in a suitable manner so that the resulting topology is 6-regular with n^6 processors and a diameter of only *3n*. We have shown that the connectivity of this network is 6 and the diameter is only O ($N^{1/6}$) in contrast to O ($N^{1/3}$) on a 3-dimensional torus with the same node degree of 6. In this thesis, we have proposed an optical implementation for the inter-block connections of the Multi-Mesh, where we use the advantages of wavelength division multiplexing (WDM).

For effective use in parallel processing, it is essential that the delay along each link is small and uniform (O (1)). Since the inter-block links used in the 3D MM are relatively long, optical links for such inter-block connections may be used to ensure a small uniform delay link. The intra-block links, however, can always be kept electronic since they introduce short links of constant length. In recent years, the optical interconnection system has also drawn much attention among researchers because of its superior power, speed and crosstalk properties compared to the electronic links when the interconnection distance is more than a few millimeters [ChKr93], [LoSu94b].

1.5 Thesis Organization

In chapter 2, we provide a brief overview of the related fields of our research. First of all we discuss the concept of multiprocessor architecture, interconnection networks, network parameters, and some examples of static multiprocessor architectures. Then we present some optical technology and optical network components that are used to optical implementation of the hybrid networks and some examples of hybrid networks.

In chapter 3 we describe our proposed 3D MM network topology, studied its various topological properties and provide a table of comparison to compare the topological properties of our proposed network with other similar networks. In this chapter we also discuss the communication algorithm for routing on the 3D MM network and the fundamental algorithm on 3D MM network.

In chapter 4, we propose how the Multi-Mesh architecture may be implemented using optical technology and we describe a number of possible approaches for designing optics-based interconnections for the Multi-Mesh.

We provide summary of works, proposed possible future directions and concluded in chapter 5. In Appendix A we give some more path calculations for different source and destinations and in Appendix B we provide glossary of important terms. Finally we give the bibliography.

Chapter 2

Literature Review

In this chapter we have provided a brief overview of the related fields of our research. First of all we discuss the concept of multiprocessor architecture, interconnection networks, network parameters, and some examples of static multiprocessor architectures. Then we present some optical components that are used to optical implementation of the hybrid networks and some examples of hybrid networks that are similar to our proposed network.

2.1 Multiprocessor Architecture

A multiprocessor architecture and/or distributed computer consists of a number of processing units or computers that are also called nodes that work simultaneously and/or independently to solve a given task. A fundamental problem in any multiprocessor system is to maintain an efficient data communication between the processors of the network. In a multi-processor network each node consists of a data processor (DP) executes and a communication processor (CP) [Zo96] as shown in Figure 1.1(b). The data processor executes algorithms and the communication processor is responsible for routing and point-to-point communication mechanism shown in Figure1.1 (a). In the network, the hardware that is used to move the messages is known as routers that are situated in the CP.

Communication processors also contain a buffer that contains the messages to be sent to the next node. To get full parallelism, nodes are required to get the data to the right place within a reasonable amount of time [Le92]. Data communication and sharing occurs by sending messages to each other. Since a node is not directly connected to all other nodes, a message is moved from one node to another until the message reaches its destination [To94].

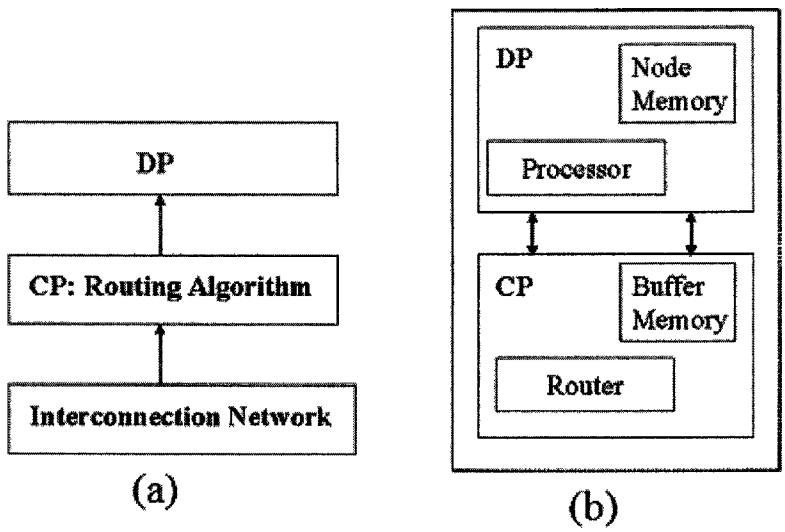

Figure 2.1: (a) Communication between CP and DP, (b) A node in a network

2.2 Interconnection Network

In a multi-computer architecture, there can be situations where thousands of processing units work simultaneously to solve a given problem. These processors may need to share data or to send messages to each other. Since the processors are not directly connected to all other processors, it is important to ensure that any processor may communicate with any other processor simply and quickly. An important requirement of an interconnection network is that any pair of processors should be able to communicate with each other as fast as possible.

2.3 Network Parameters

Interconnection networks are characterized by a number of parameters. Some of the most important parameters are given below-

- **Network size:** Total number of nodes in a network

- **Node degree:** The degree of a node is the total number of incoming and outgoing links [Be73]. The node degree represents the cost of a node from the communication point of view and hence a network topology with fixed and low node degree is favorable [SFK97].

- **Diameter:** The diameter of a graph G is the maximum of the shortest distance (hops) between any two nodes [Be73]. For a multi-processor architecture, the diameter is an important attribute and is related to information transfer delay. In order to achieve faster data communication, diameter should be kept as small as possible.

- **Connectivity:** Connectivity is the minimum number of arcs that have to be removed from the network to cut the network into two disconnected networks [Be73], [SFK97]. A graph with a connectivity of C can tolerate up to C-1 edge-faults, since any pair of fault-free nodes can still find a path between the fault-free nodes. In other words, a network with a higher connectivity is preferable from the point of view of fault tolerance.

- **Cost:** The total number of communication links required by the network defines the network cost [SFK97].

2.4 Types of Interconnection Network

There are two basic types of interconnection network, static and dynamic.

- **Static interconnection network**

In case of a static interconnection network, all connections among the processors are fixed meaning that the processors are wired directly [SFK97]. Static interconnection networks are better where the problems are uniform and the communication pattern is predictable [SFK97].

- **Dynamic interconnection network**

The connections between the processors can be changed as the processors are connected by switch instead of direct wire. Dynamic interconnection is expensive.

Here we are only interested about the static interconnection network.

Static interconnection topologies

The way that nodes are interconnected is called the network topology [To94]. Static interconnection topologies can be classified according to their dimensions [SFK97]:

- One-dimensional topologies
- Two – dimensional topologies
- Three-dimensional topologies
- Multidimensional topologies e.g. Hypercube, De Bruijn, Kautz etc.

Figure 2.2 [SFK97] shows the classification of static interconnection topology-

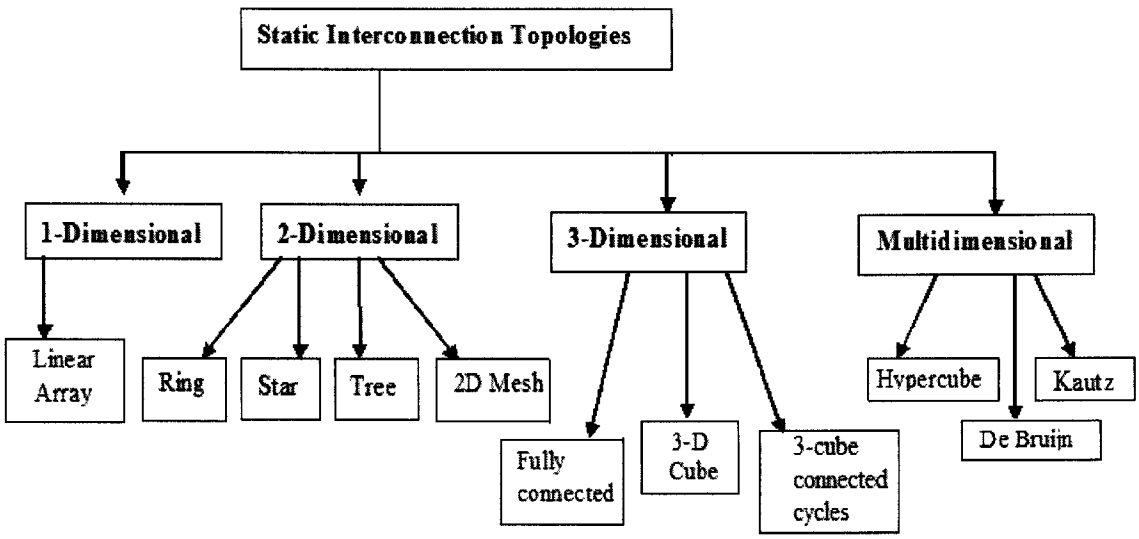

Figure 2.2: Types of interconnection topologies

2.5 Examples of Some Simple Multiprocessor Architecture

2.5.1 Linear array

Linear array is the simplest and cheapest way to connect the processors of a parallel computer. Each processor has direct connection with two other processors except the boundary processors that have one. Figure 2.3 shows an example of such network. This network topology has worst diameter that is n-1 and arc connectivity that is only 1 [SFK97].

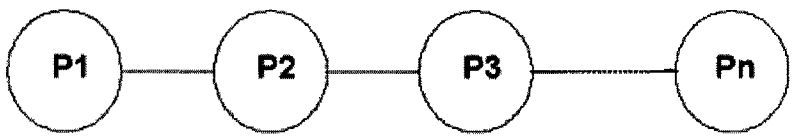

Figure 2.3: Linear array

2.5.2 Ring

If the boundary nodes of a linear array are connected to each other, then the network topology is called ring topology. Figure 2.4 shows the ring network from a linear array. In a ring network all the nodes have two connections. It improves the connectivity and diameter of the linear array by a factor of two: diameter = n/2 and arc connectivity =2.

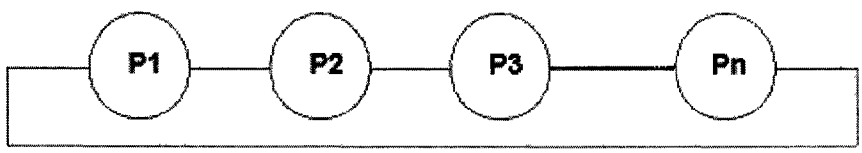

Figure 2.4: Ring

2.5.3 Star

In a star topology there is one central node, to which all other nodes are connected as shown in Figure 2.5. Central node has n – 1 connections where all other nodes have only one connection.

Star network is a simple topology but not suitable for large configuration, as the number of connections increases for the center node with the increase of nodes.

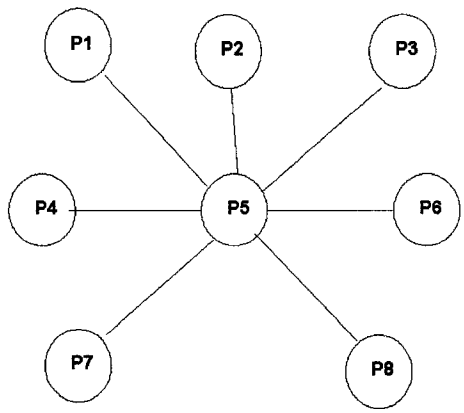

Figure 2.5: Star network

2.5.4 Tree

Nodes are interconnected in a tree structure as shown in Figure 2.6. It has smaller diameter (log n) and the degree of the nodes are 1 for leaf nodes, 2 for root node and internal nodes have 3. The main drawback of this network is its poor arc connectivity that is only 1.

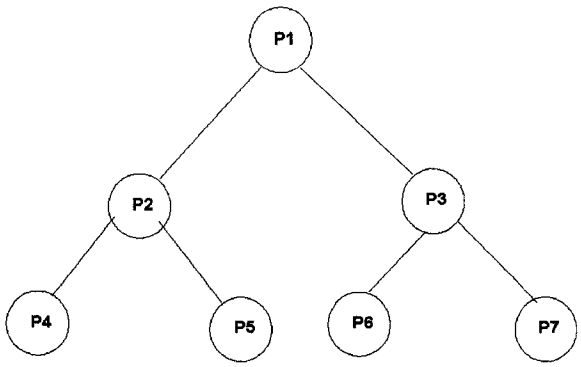

Figure 2.6: Tree interconnection network

2.5.5 Fully connected network

In this network topology, all the nodes are directly connected to each other. This topology is ideal from the point of view of network diameter that is 1 but node degree is n-1 for all the nodes. So the cost of this network is extremely high and it is not scalable to massive parallel computer.

The following figure shows an example of a fully connected network.

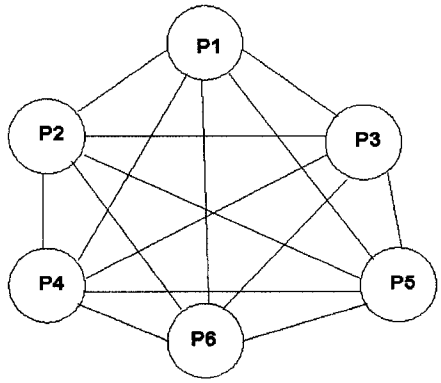

Figure 2.7: Fully connected network

2.5.6 Hypercube

One of the most popular topologies is the hypercube topology. The hypercube is one example of a multidimensional mesh of processors. A d-dimensional binary hypercube has 2^d nodes. The hypercube topology is attractive for its small diameter (log n) and arc connectivity (log n). In binary hypercube each node has an address - a number between 0 and $2^d - 1$. Two processors whose binary representations differ in exactly one bit are connected together. This property greatly facilitates the routing of messages through the network [LoSu94a]. The major disadvantage of hypercube is that its node degree is log n and hence the node degree grows as n increases [SFK97].

For a d-dimension hypercube, each node is connected to d nodes. Figure 2.8 shows an example of hypercube network. The hypercube and other related networks suffer from lack of scalability [LoSu94a].

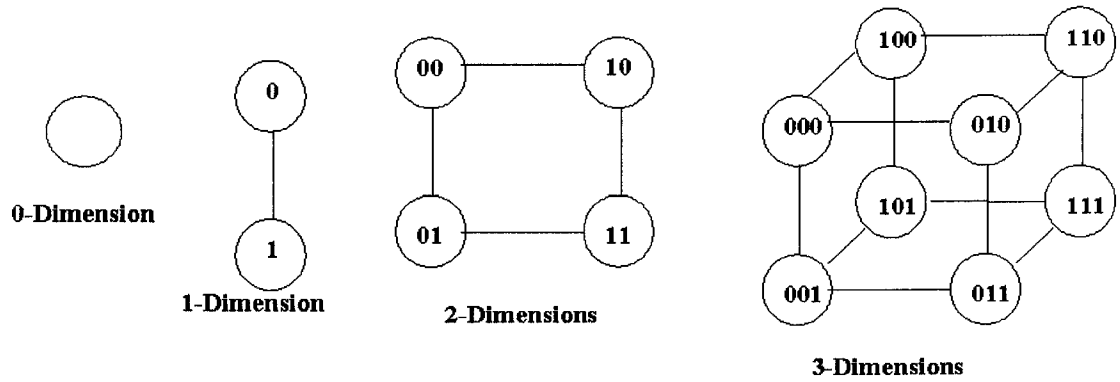

Figure 2.8: Hypercube topologies of different dimensions

2.5.7 Mesh network

The term mesh has been used by various investigators in different ways. Following Ullman, we will use the term mesh to denote a square grid of processors [Ul84], so that the mesh network is a two dimensional arrangement of nodes in a Manhattan Street architecture. Among the static interconnection networks, the two-dimensional mesh is one of the most popular architectures as it has a very regular and simple architecture [HwBr83], [St83], [Le92]. Due to the constant node degree, the mesh network is highly scalable [LoSu94a]. Researchers have proposed several variations of the mesh architecture using techniques such as wrap-around, diagonal interconnections among the nodes. Some popular networks based on the mesh architecture are called two dimensional (2D) mesh, two-dimensional wrap around mesh (also called Torus) and three-dimensional (3D) mesh.

2-D Mesh

The most popular mesh is the so-called two-dimensional (2D) mesh where nodes are arranged in a grid pattern as shown in Figure 2.9 [Ra92]. Except for the boundary processors every other processor is connected to its neighbours to the left, right, above and below through bi-directional links [Ra92]. Mesh networks represent a good compromise among the contradictory requirements of static network parameters [SFK97]. It has, relatively speaking, a short diameter and arc connectivity [SFK97]. In a 2-dimensional mesh, it is convenient to identify nodes by the x-y coordinate values of their positions. Meshes are easy to implement and extend. Variations of the mesh topology are possible, depending on whether there is any wrap-around or diagonal interconnections among the nodes [SFK97].

Figure 2.9(a) and 2.9(b) show two different types of two-dimensional mesh network.

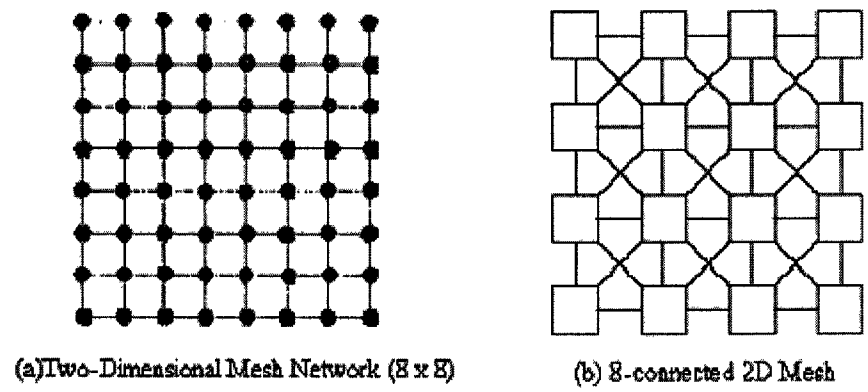

Figure 2.9: (a) Two-Dimensional mesh (8x8), (b) 8-connected 2D mesh

Torus

A torus is defined as a mesh with wrap-around links as shown in Figure 2.10. We refer to a processor in row i and column j as $P_{i,j}$, $0 \leq i, j < n$. Processor $P_{i,0}$ is connected to $P_{i,n-1}$ and $P_{0,j}$ is connected with $P_{n-1,j}$ [SFK97]. The 2D mesh and the torus network topologies are attractive because of simplicity, regularity, scalability and efficient use of space for their VLSI layouts [Le92], [SFK97], [Ra92].

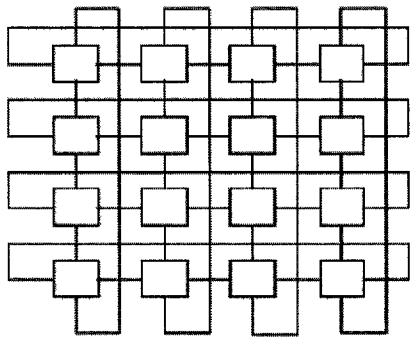

Figure 2.10: Torus Network

Multi-Mesh

The Multi-Mesh (MM) interconnection network topology was proposed by D. Das, M. De and B. P. Sinha [DDS99]. The MM has been proposed as an efficient topology for optical networks [Le92] and peer-to-peer networks. The MM topology of order n uses multiple $n \times n$(two dimensional) meshes as the basic building blocks, n^2 meshes are again arranged in the form of $n \times n$ matrix and each matrix is termed as a block. A processor inside the block can be identified by specifying its x and y coordinates in its matrix. Similarly a block can also be identified by its x and y coordinates. In our notation B (α, β)

identifies a block where α, β ($1 \leq \alpha, \beta \leq n$) are the x and y coordinates identifying the block and P (α, β, x, y) identifies a processor where the first two coordinates represent the position of its block and the last two coordinates represent the location of the processor within the block. If two processors are within the same block and are connected by an edge, we will call the two processors to be *neighbors*.

Based on the neighborhood, processors within a block are categorized into the following three classes-

1) **The processors with two neighbors** (the processors on the corners of the block) -

- $x = 1$ or $x = n$,
- $y = 1$ or $y = n$

We will call such processors as *corner processors*. It is obvious that in a 2D mesh there are exactly four such processors.

2) **The processors with three neighbors** – the processors on the sides of the block (but not on corners) are characterized by x and y values such that exactly one of these coordinates are 1 or n. Such processors have

- $(1 < x < n, y = 1$ or $y = n)$ or
- $(x = 1$ or $x = n, 1 < y < n)$.

We will call such processors as *boundary processors*. There are $4(n-2)$ such processors.

3) **The processors with four neighbors** – the processors each having four neighbors are called *internal processor*. There is exactly $(n-1)^2$ such processors in a 2D block.

D. Das et al [DDS99] have defined interconnection rules so that the proposed network topology of order n contains n^4 processors by interconnecting n^2 two dimensional meshes, each with n^2 processors.

We show a MM network order 3 in Figure 2.11 [DDS99]. We have not shown all the interconnecting for clarity.

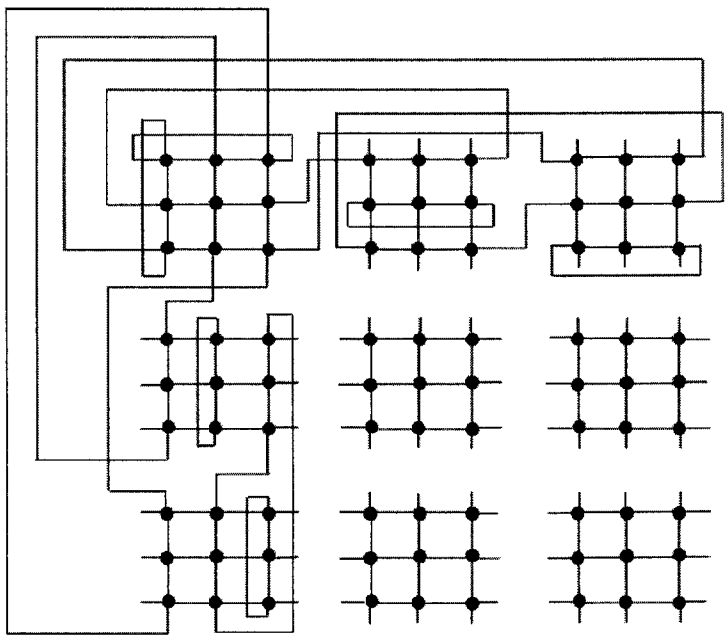

Figure 2.11: A Multi-Mesh network with 3 X 3 meshes

The Multi-Mesh topology corresponds to a regular graph[1]. With the same number of processors and the same number of links as in the case of a torus, the Multi-Mesh topology [DDS99] has the advantage of offering a much lower diameter. The authors have shown that the time complexities of different basic operations mapped on it are considerably less than those for many existing mesh-type topologies [DGS97], [DDS99]. Because of this property, it has also been proposed as an efficient topology for optical networks [SFK97].

[1] In a graph if each node has same node degree, then the graph is called a regular graph [Be73].

3D Mesh

The three-dimensional mesh or the 3D mesh can be thought of as n layers of 2D meshes arranged in the third dimension or the z direction [LiFi01]. As a result, there are $n \times n^2 = n^3$ processors in a 3D mesh. Figure 2.12 shows an example of a 3D mesh. The three dimensional (3D) mesh improves the diameter (from $N^{1/2}$ to $N^{1/3}$) and arc connectivity (from 2 to 3) compared to the 2D mesh.

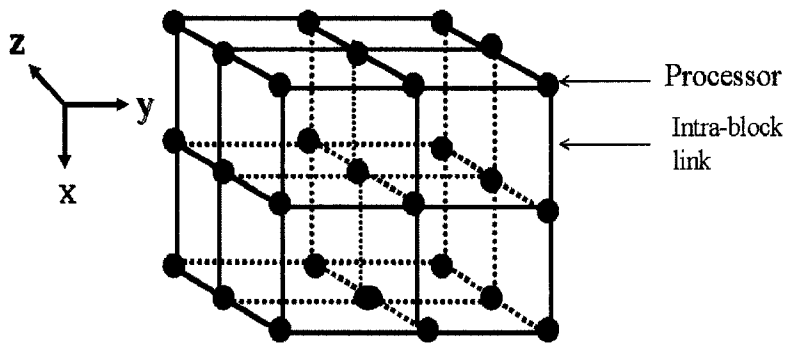

Figure 2.12 3D mesh

In a 3D mesh, n^3 processors are arranged along three orthogonal dimensions, say x, y and z, so that a processor at coordinates (x, y, z) (which we will denote by P(x, y, z)) is connected to six other neighbouring processors at P(x+1, y, z), P(x-1, y, z), P(x, y+1, y, z), P(x, y-1, z), P(x, y, z+1) and P(x, y, z-1), when they exist, for all integer values of x, y and z, $1 \leq x, y, z \leq n$. Some processors have 3, 4 or 5 neighbours depending on their position in the 3D-mesh while the remaining processors have all 6 neighbours.

2.6 Optical Technology and Optical Communication

The primary bottleneck in today's metal-based interconnection networks is the very limited bandwidth of long copper lines, which results in limited communication speed [LoSu94b]. Optical interconnects offer high-speed computers key advantages over metal interconnects which includes: (1) high spatial and temporal bandwidths, (2) high-speed transmission, (3) low crosstalk independent of data rates, and (4) high interconnect densities [LoSu94b].

In this section, we will describe the following topics on optical devices and optical networking related to our investigation:

- Optical fiber
- Optical couplers
- Routers
- WDM networks
- Wavelength routed Network
- Use of optical technology in interconnection network design

Due to lack of space we will not review details of optical technology such as optical amplifiers, receivers and transmitters, filters and gratings [So03], [Mu97], [StBa99], [BBRM97].

2.6.1 Optical fiber

Optical fiber is the medium of data transmission in an optical network. Optical fiber is a thin filament of glass, which acts as a wave-guide [BBRM97]. Fiber is attractive as a communication medium due to the following advantages [So03], [Mu97]:

- High speed,
- Huge bandwidth
- High security
- Low bit error rate,
- No electromagnetic interference,
- Low power requirement and
- Low signal attenuation.

2.6.2 Optical couplers

Coupler is a general term that covers all devices that combine beams of light into or split into beams of light out of a fiber [Mu97]. A splitter is a coupler that divides the optical signal on one fiber to two or more fibers. Combiners are the reverse of splitters, and when turned around, a combiner can be used as a splitter [Mu97]. The following figure [Mu97] shows an example of these devices-

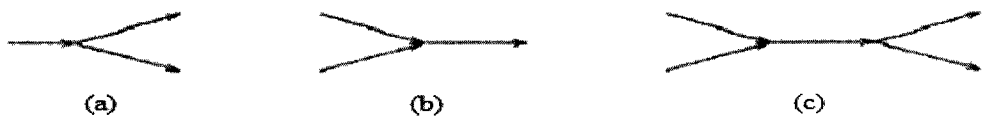

Figure 2.13: (a) Splitter, (b) Combiner and (c) Coupler

2.6.3 Optical multiplexers and demultiplexers

Optical multiplexers are used to combine several independent signals at different wavelength into one fiber. A demultiplexer works exactly the opposite way that is splitting the signals at different wavelengths. Figure 2.14 shows an example of a multiplexer and a demultiplexer.

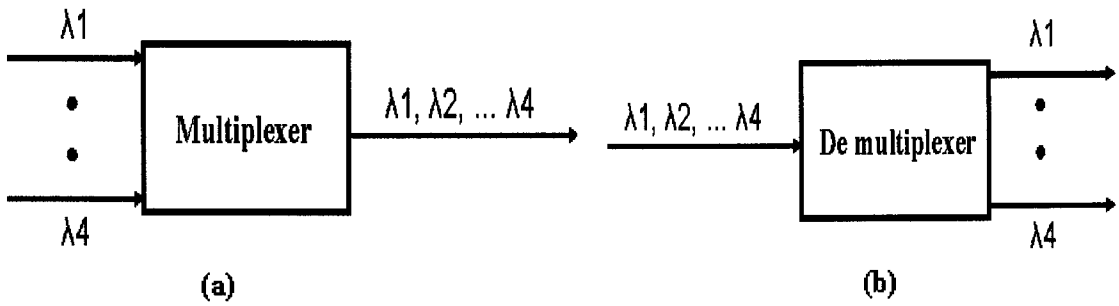

Figure 2.14: (a) Multiplexer (b) Demultiplexer

2.6.4 Passive star coupler

One type of optical networks using multi-wavelength fiber links is to use a passive star coupler, the star coupler is a "broadcast" device, so that an optical signal transmitted using a given wavelength from a node in the network will be communicated to all other nodes in the network. This means that the power of the transmitted signal will be equally divided among all the output ports connected to the coupler [Mu97]. Figure 2.15 [Mu97] shows an example of a passive star coupler where a signal using wavelength λ1 from input fiber 1 and another on wavelength λ4 from input fiber 4 are broadcast to all output

ports. There is a problem in using star coupler that is, a collision may occur when two or more signals from the input fibers are simultaneously launched into the star on the same wavelength [Mu97].

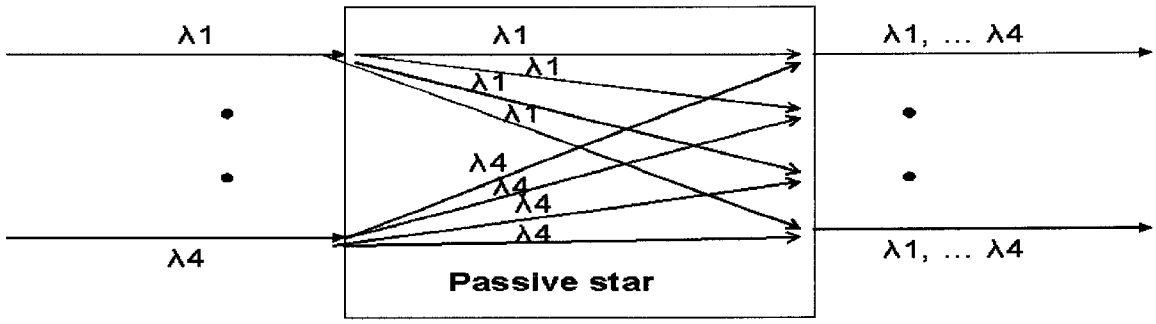

Figure 2.15: A 4 x 4 passive star coupler

2.6.5 Routers

In an optical network, a router is a device that is connected to a number of fibers, some carrying incoming optical signals to the router and the others carrying outgoing optical signals. A router determines how the incoming signals will be directed to outgoing fibers. Figure 2.16 shows a router with 3 fibers carrying incoming signals and 3 fibers carrying outgoing signals.

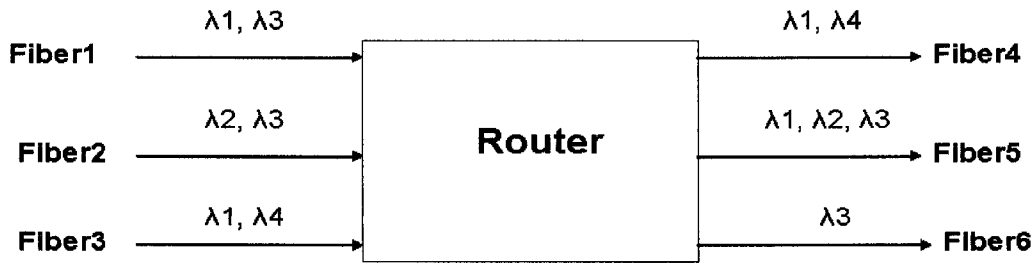

Figure 2.16: Router

The control settings on the router determine the actual routing. For example in figure 2.16, the signal at wavelength λ_1 on fiber 1 may need to be directed to fiber 5. Figure 2.17 [Mu97] shows an example of a passive router – wavelengths λ1, λ2, λ3 and λ4 incident on Input fibers 1, 2, 3 and 4 respectively [Mu97]. By using this device we can reuse the wavelengths.

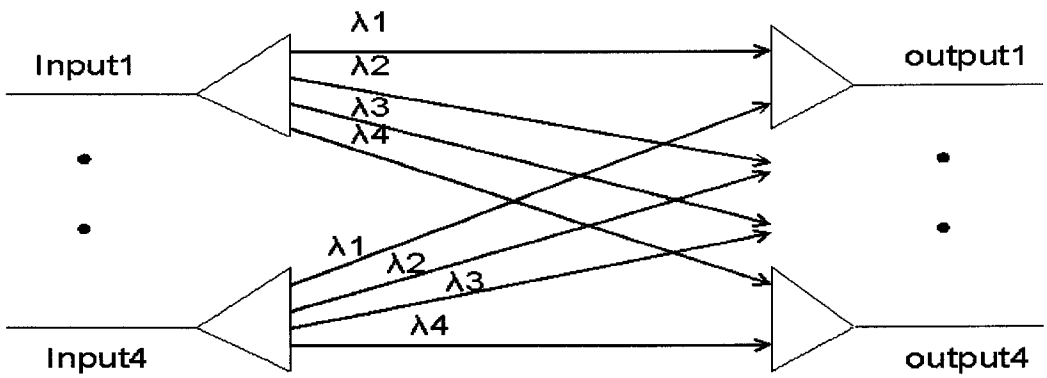

Figure 2.17: A 4 x 4 passive Router

Figure 2.17 shows how a number of MUX/DEMUX allows us to define routers.

2.6.6 WDM network

The huge bandwidth of optical fiber allows a tremendous amount of data transmission rate. It is technologically impossible to exploit all of that bandwidth using a single high-capacity channel [StBa99] . Due to the fact that this is enormously more than the speed of electronic communication, *wavelength-division multiplexing* (WDM) is a promising approach that can be used to exploit the huge bandwidth of optical fiber [Mu97] [BBRM97] , [StBa99]. In WDM, the optical transmission spectrum is divided into a

number of non-overlapping wavelength (or frequency) bands, with each wavelength supporting a single communication channel operating at peak electronic speed [BBRM97].

2.6.7 Wavelength routed network

A WDM network using passive coupler is not viable when the network contains a large number of nodes due to the power requirements of such a broadcasting network [Mu97]. A wavelength routed WDM network is a network where each end-node (the source or destination of data) is connected to a router and each router is connected to other routers. Figure 2.18 shows a small wavelength routed WDM network where a square represents an end node and oval represents a router. The advantage of such network is that the data is not broadcast to all the end-nodes. The settings of the routers determine which end-nodes will be connected by a lightpath (all-optical path through which the information flows in a wavelength-routed optical network).

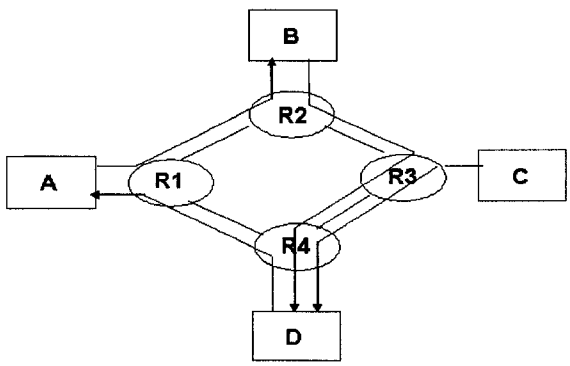

Figure 2.18: A wavelength-routed network

2.6.8 Single-hop network

A network in which a packet is sent directly (in one hop) from it's source processor to the destination processor without routing through any intermediate processor [Mu97].

2.6.9 Multi-hop network

A network in which a packet may travel through zero or more intermediate processors before it reaches to its final destination [Mu97].

2.6.10 Routing and wavelength assignment

Given a network topology and a set of lightpaths (to be determined), routing the lightpaths in the network and assigning wavelengths to these lightpaths is referred as the routing and wavelength assignment (RWA) problem [Mu97].

2.6.11 Fault tolerant optical network

With WDM optical network each physical fiber link is able to support many lightpaths. As network grows in size and complexity the amount of lightpaths become more, so the failure of a fiber link may causes to significant data losses. In order to have a fault tolerant WDM network, it is very important to handle these types of fiber faults. Since single fiber failures are the major form of failures in optical networks, in this thesis our focus is on the single faults.

2.7 Use of Optical Technology in Interconnection Network Design

In realizing VLSI multiprocessor systems, an obvious approach for creating the links between processors is the use of VLSI fabrication technology for example using the metal1 or metal2 layer and has been done for some time [Ul84]. It is well known that implementing copper base connections to realize complex interconnection is problematic since long copper wires are needed for such complex topologies [Ul84]. A problem of metal interconnect technology is that long copper wire accentuates problems like skin effect, crosstalk, interference, wave reflections, electrical noise due to current changes, and dielectric imperfections [LoSu94b]. These problems can cause severe pulse distortions and attenuation, clock skew, and random propagation delays [StCo91].

According to Louri and Sung [LoSu94b] multiprocessor systems based on metal interconnects experience the technological limitations of communication bandwidth constraints, low interconnect density, long network latencies, and high power requirements. Metal-based communications between subsystems and chip has become the limiting factor in high-speed computing; maturing optics-based technologies offer advantages that may unplug this bottleneck [LoSu94b].

As optical technology has evolved in the last decade, an obvious approach to this bottleneck is to use optical technology.

2.7.1 Advantages of optical interconnects in multi-computers systems

A summary of advantages of using optical interconnections are given below [LoSu94b], [ChKr93], [So03]:

- Optics allows inherent parallelism
- Optical Communication has higher bandwidth
- Optical signal propagate in parallel channels without interference
- There is less signal crosstalk in optical communication
- Optical communication is inherently immune from electromagnetic interference and ground loops
- There is lower signal and clock skew and lower power dissipation in optical communication
- Propagation speed for optical signals is, for short distances, essentially independent of communication distance
- There is potential for reconfigurable interconnects

2.7.2 Free space optical interconnects

Free-space optical interconnects exploits air space for optical signal propagation [LoSu94b]. In order to provide communication channel for free space interconnection, lenses and holograms are used as optical elements.

Free space interconnects are classified into two categories [LoSu94b]:

1. Space-variant
2. Space-invariant

A totally space-invariant network has a regular structure where each node has same connection patterns shown in Figure 2.21 [LoSu94b] whereas in totally space-variant

network there is no regular pattern (arbitrary interconnection) between the nodes. Figure 2.21 shows such interconnection.

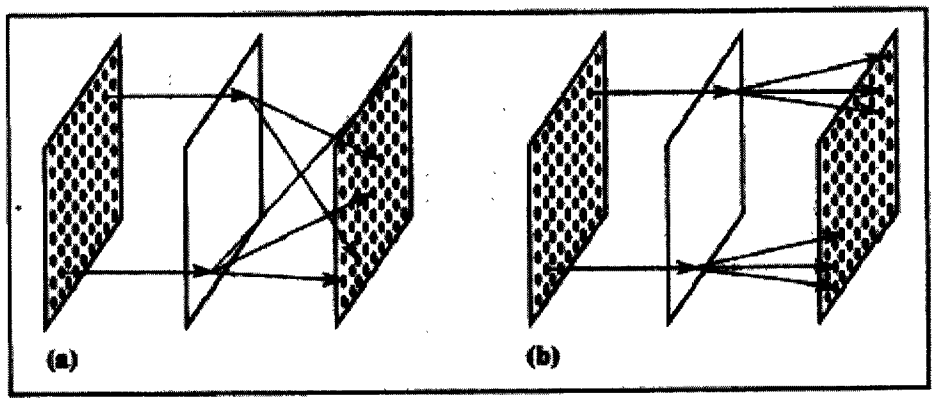

Figure 2.19: Free-Space: (a) Space-Variant (b) Space-Invariant

2.8 Interconnection Networks based on Opto-Electronic Technology

In this section we present two recently proposed high-throughput hybrid optical multiprocessor architectures.

2.8.1 OMMH

Optical multi-mesh hypercube (OMMH) network topology for multiprocessor network is proposed by Louri and Sung [LoSu94a]. The OMMH network uses meshes and hypercube as the basic building blocks. This network topology combines the advantages of meshes (constant node degree and scalability) and hypercubes (small diameter, high connectivity, symmetry, simple control, routing and fault tolerance) and avoids the disadvantages of the lack of scalability of hypercube and the large diameters of meshes.

This network can maintain constant node degree regardless of the increase in the network size [LoSu94a]. The authors claim that the flexibility of the OMMH network makes it well suited for optical implementations. The OMMH network uses a three dimensional optical design based on free-space optics. The analysis and simulations results in [LoSu94a] show that the OMMH network is scalable, efficient in communication and highly fault-tolerant. Optical implementation of the network is possible with the existing hardware. Figure 2.22 [LoSu94a] shows an example of an OMMH network.

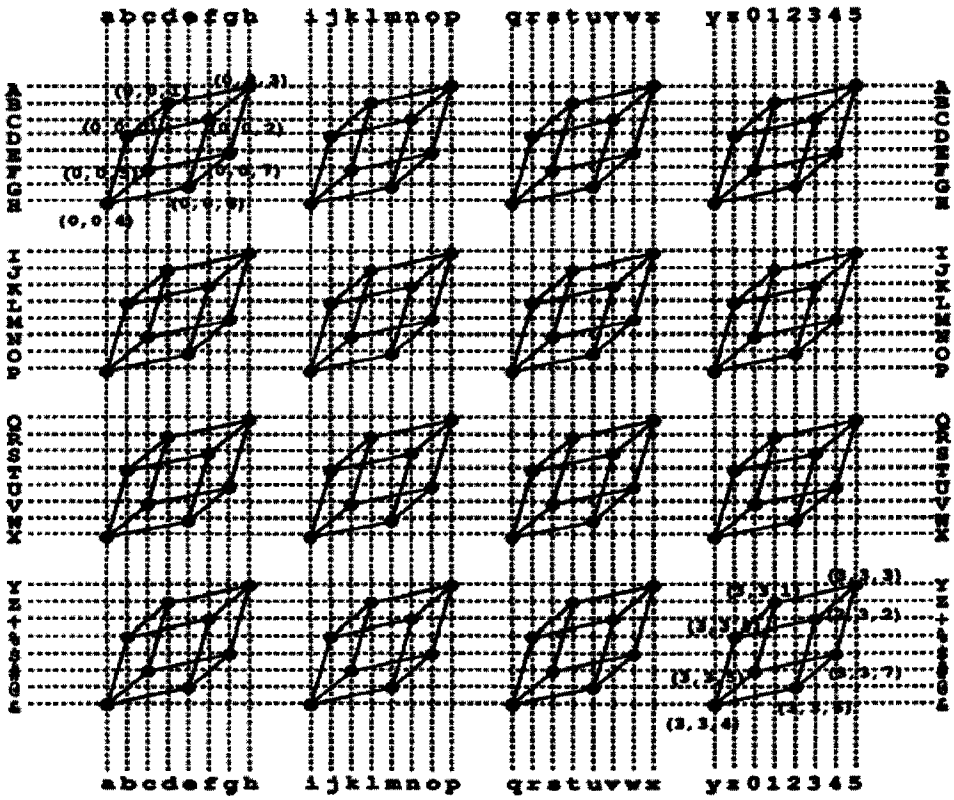

Figure 2.20: A (4, 4, 3) OMMH network with 128 nodes

2.8.2 OTIS-Mesh

Optical Transpose Interconnect System (OTIS) is proposed by Marsden et. al [MMHE93]. OTIS architecture is an example of a hybrid architecture in which the processors are partitioned into groups where processors within a group are connected by electronic links and processors situated on the different groups are interconnected by optical links.

OTIS-Mesh is a type of OTIS computers where a number of well known algorithms can be efficiently mapped on OTIS-Mesh architecture [Os00], [SaWa97], [WaSa00], [ZMPE00]. OTIS-Mesh is also a hybrid architecture that uses the same idea of OTIS computer. Figure 2.21 [WaSa01] shows an OTIS-Mesh containing 16 processors where small square boxes denote processor and large square boxes represents a group of processors. The groups are arranged in two-dimensional arrays.

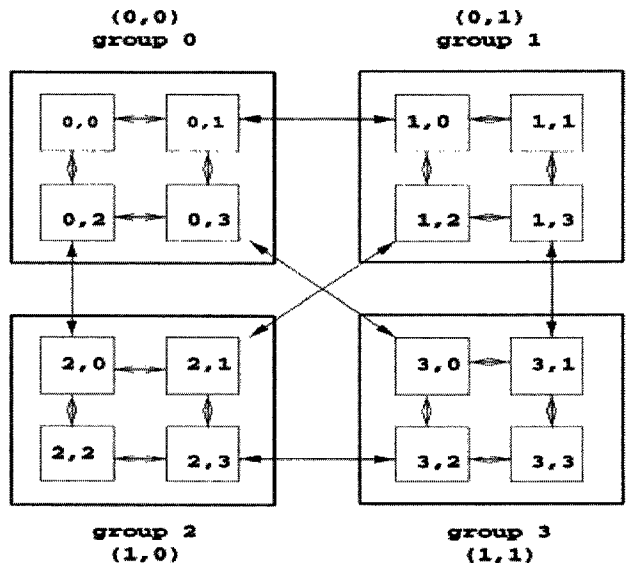

Figure 2.21: An example of OTIS-mesh network with 16 nodes

Chapter 3

Topology of 3D Multi-Mesh

In this chapter we introduce a new network topology, called the 3D Multi-Mesh (3D MM) for multiprocessor architecture. As discussed in chapter 2, the 2D Multi-Mesh architecture [DDS99] uses a $n \times n$ mesh of processors as its basic building block and each processor in a $n \times n$ mesh may be identified by specifying a x-coordinate value (say x) and a y-coordinate value (say y). In a block, the processors having x = 1, x = n or having y = 1 or y = n have less than 4 connections to other processors within the same block. We have seen that these processors are connected to processors in other blocks of $n \times n$ processors in a particular pattern resulting in a network with attractive topological properties. It is well known that a $n \times n \times n$ mesh has better diameter and connectivity compared to a $n \times n$ mesh [LiFi01], [SFK97]. It is therefore reasonable to extend the idea of interconnecting blocks of 2-dimensional (i.e., $n \times n$) meshes of processors to the idea of interconnecting 3-dimensional (i.e., $n \times n \times n$) meshes of processors. This is the topic that we will explore in this chapter. Our proposed network consists of n^3 three-dimensional meshes, each having n^3 processors, interconnected in a suitable manner so that the resulting topology is 6-regular with n^6 processors. We will call such a network a 3D Multi-Mesh (3D MM) of order n. In this chapter we introduce the 3D MM topology, analyze its architectural properties and compare it to other network architectures.

3.1 Description of a 3D Block

The basic building block of the proposed 3D MM of order n is the $n \times n \times n$ Mesh, which we will call a *3D block*, consists of n^3 processor nodes. We may visualize a 3D block of n^3 processors as consisting of n planes of $n \times n$ two-dimensional meshes of processors. We show an example of a 3D block of order 3 ($n = 3$) in Figure 3.1. The 3D MM of order n consists of n^3 such 3D blocks arranged in a three-dimensional $n \times n \times n$ array, so that there are altogether $N = n^6$ processors in a 3D MM network. We show a 3D MM network of order 3 in Figure 3.2. An $n \times n \times n$ 3D block has (n-2) × (n-2) × (n-2) processors in the block, each having 6 links to other processors inside the same block. Each of the remaining processors lies on the six faces[1] of the block and has 3, 4 or 5 links, depending on the position of the processor in the block. Extending the idea used in the Multi-Mesh [DDS99] architecture, we connect the processors on the six surfaces of a 3D block to the processors on the faces of other 3D blocks according to the inter-block rules described in the section 3.1.3.

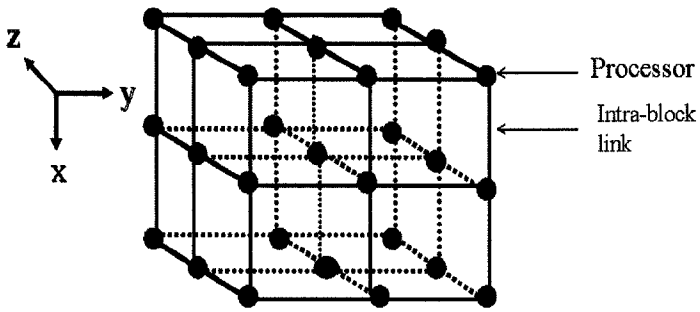

Figure 3.1: A 3D block of order 3

[1] A face of a cube represents the first or the last plane of 3D mesh. A processor P(x, y, z) on the face of a cube have the value of 1 or n for at least one of the coordinates x, y or z.

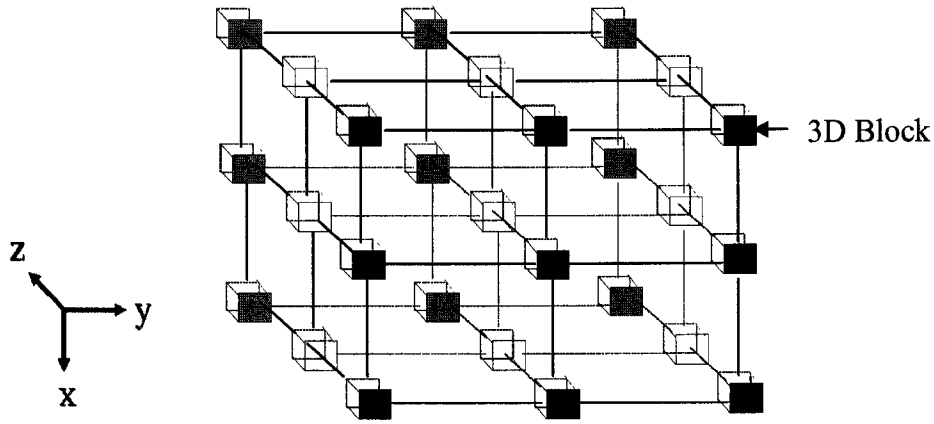

Figure 3.2: 3D MM network of order 3

3.1.1 Intra-block connection

We arrange the three dimensional mesh (forming the basic building block of the proposed 3D Multi-Mesh network) consisting of n^3 processors along the three orthogonal dimensions, say x, y and z, so that a processor within a 3D block is uniquely identified by three coordinates x, y, z. A processor identified by the coordinates x, y, z is connected to six other *neighboring processors* (processors within the block that are connected by links), when they exist. These neighboring processors are identified by –

- (x + 1, y, z),
- (x - 1, y, z),
- (x, y +1, z),
- (x, y-1, z),
- (x, y, z+1) and
- (x, y, z-1)

Figure 3.1 shows how neighboring processors are connected by the *intra-block links*.

3.1.2 Categorization of processors

It is important to note that all 6 neighboring processors may not always exist. We will term a processor, which has all of its six neighbors in all three dimensions (for $1 < x, y, z < n$), as an *internal processor* since all its connections are to other processors within the same block. However, the processors on the six faces of the block identified by $x = 1$, $x = n$, $y = 1$, $y = n$, $z = 1$ and $z = n$, will have less than six neighbors each. We categorize these processors as follows:

1) **The processors with three neighbors** – the processors on the corners of the block have

- $x = 1$ or $x = n$,
- $y = 1$ or $y = n$,
- $z = 1$ or $z = n$.

We will call such processors as *corner processors*. It is obvious that we will have exactly eight such processors.

2) **The processors with four neighbors** – the processors on the sides of the block (but not on corners) are characterized by x, y and z values such that exactly two of these coordinates are 1 or n. Such processors have

- ($x = 1$ or $x = n$, $y = 1$ or $y = n$, $1 < z < n$) or
- ($x = 1$ or $x = n$, $1 < y < n$, $z = 1$ or $z = n$) or
- ($1 < x < n$, $y = 1$ or $y = n$, $z = 1$ or $z = n$).

We will call such processors as *boundary edge processors*. We will have exactly $8(n-2)$ such processors.

3) **The processors with five neighbors** – the processors on the faces of the block (but not on sides or corners) are characterized by x, y and z values such that exactly one of these coordinates is either 1 or n. Such processors have

- $(x = 1 \text{ or } x = n, 1 < y < n, 1 < z < n)$ or
- $(1 < x < n, y = 1 \text{ or } y = n, 1 < z < n)$ or
- $(1 < x < n, 1 < y < n, z = 1 \text{ or } z = n)$.

We will call such processors as *face-centered processors*. We will have exactly $6(n-2)^2$ such processors in a 3D block.

3.1.3 Inter-block connections

Our 3D Multi-Mesh is an interconnection of n^3 3D blocks arranged along the three orthogonal dimensions as shown in Figure 3.2. We designate with the symbols α, β and γ respectively (to make them distinct from x, y and z) the coordinate values along the three orthogonal dimensions. Thus, we now have a total of n^6 processors where each processor can be uniquely identified by its six coordinate values α, β, γ, x, y, z that we will denoted by P (α, β, γ, x, y, z). We will characterize any particular 3D block by a given set of values for α, β and γ coordinates and we will denote a block by B (α, β, γ). We connect all the processors on the six faces of each 3D block to the processors on the faces of other 3D blocks by one or more *inter-block links* so that each processor eventually has exactly six links to other processors (either in the same 3D block or in other 3D block(s)).

We describe below the rules for the inter-block links.

3.1.4 Rules for inter-block connections

Inter-block Rule 1: (Links from y = 1 and y = *n* planes)

The processor P (α, β, γ, **x**, 1, z) is connected to the processor P (α, **x**, γ, β, *n*, z) by a symmetric link for all α, γ, z where $1 \leq \alpha, \beta, \gamma, x, z \leq n$. We denote this by

$$\forall \alpha, \gamma, z, P(\alpha, \beta, \gamma, \mathbf{x}, 1, z) \leftrightarrow P(\alpha, \mathbf{x}, \gamma, \beta, n, z)$$

Such links allow us to interchange only the values of β and x and we will refer to these links using the notation $\forall \alpha, \gamma, z\ (\beta \leftrightarrow \mathbf{x})$. We note that the value of z is not changed for the processors connected by these links.

Inter-block Rule 2: (Links from x = 1 and x = *n* planes)

The processor, P (α, β, γ, 1, y, **z**) is connected to the processor P (**z**, β, γ, *n*, y, α) by a symmetric link for all $\forall \beta, \gamma$, y where $1 \leq \alpha, \beta, \gamma, y, z \leq n$. We denote this by

$$\forall \beta, \gamma, y, P(\alpha, \beta, \gamma, 1, y, \mathbf{z}) \leftrightarrow P(\mathbf{z}, \beta, \gamma, n, y, \alpha)$$

Such links allow us to interchange only the α and z values and we will refer to these links using the notation $\forall \beta, \gamma, y\ (\alpha \leftrightarrow \mathbf{z})$. We note that the value of y is not changed for the processors connected by these links.

Inter-block Rule 3: (Links from z = 1 and z = *n* planes)

The processor P (α, β, γ, x, **y**, 1) is connected to the processor P (α, β, **y**, x, γ, *n*) by a symmetric link for all α, β, x, where $1 \leq \alpha, \beta, \gamma, x, y \leq n$. We denote this by

$$\forall \alpha, \beta, x, P(\alpha, \beta, \gamma, x, \mathbf{y}, 1) \leftrightarrow P(\alpha, \beta, \mathbf{y}, x, \gamma, n)$$

Such links allow us to interchange only the γ and y values and we will refer to these links using the notation $\forall \alpha, \beta, x\ (\gamma \leftrightarrow \mathbf{y})$. We note that the value of x is not changed for the processors connected by these links.

From the inter-block connection rules, the following properties follow immediately.

Property 1:

Starting from a given 3D block, identified by coordinates (α1, β1, γ1), we can always find a suitable processor on one of its faces, from which we can reach, using only one inter-block link, any other 3D block, identified by (α2, β2, γ2), provided exactly 2 of the coordinates of (α1, β1, γ1) are identical to the corresponding coordinates of (α2, β2, γ2).

Property 2:

The 3D Multi-Mesh corresponds to a regular graph where each processor is connected to exactly 6 other processors.

The connections are somewhat complicated; to simplify the situation, in Figure 3.3, we are showing only the blocks having $\alpha = 1$ and $\gamma = 1$ and we show only the inter-block connections along the y-axis for the processors having $z = 1$.

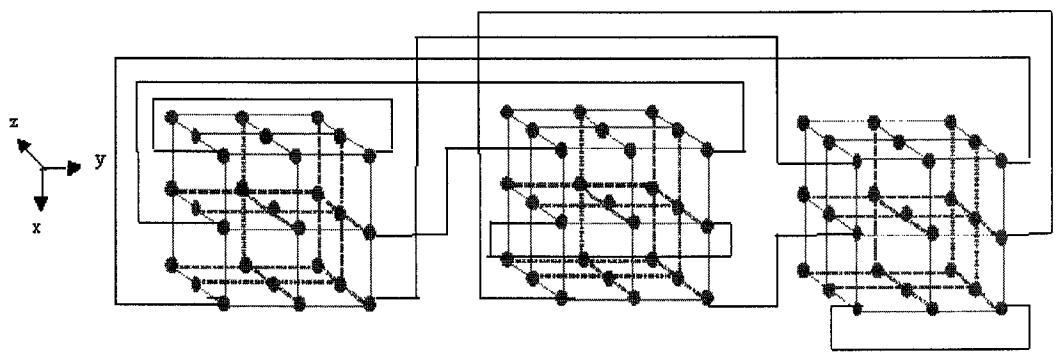

Figure 3.3 Interconnections along the y-coordinate

In Figure 3.4 we show an example of a 3D MM network of order 2 (n = 2), where we show all the inter-block connections for the processors in the block having α=1 , β = 1, γ = 1. All other links are not shown.

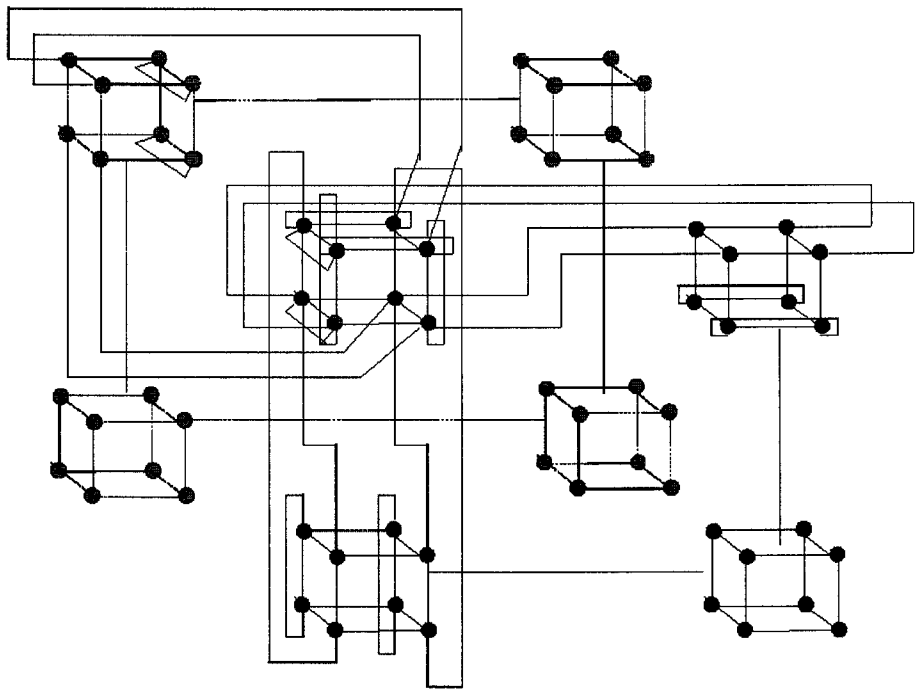

Figure 3.4: 3D MM network of order 2

3.2 Topological Properties of the 3D Multi-Mesh Network

3.2.1 Diameter

In a graph G, the diameter is the maximum possible value of the length of the shortest path between any two nodes of G [Be73]. This is a very important metric for any interconnection network. In this section we show that the diameter of the 3D Multi-Mesh of order n is $3n$. To prove this we have to show that, in a 3D Multi-Mesh of order n, it is

always possible to define a path from any source to any destination having a length of $3n$ or less.

We consider the source processor **S** = P ($\alpha 1$, $\beta 1$, $\gamma 1$, x1, y1, z1) and the destination processor **D** = P ($\alpha 2$, $\beta 2$, $\gamma 2$, x2, y2, z2), $1 \leq \alpha 1, \beta 1, \gamma 1, x1, y1, z1, \alpha 2, \beta 2, \gamma 2, x2, y2, z2 \leq n$, so that the 3D block corresponding to **S** is B ($\alpha 1$, $\beta 1$, $\gamma 1$) and that corresponding to **D** is B ($\alpha 2$, $\beta 2$, $\gamma 2$). There are three situations to consider:

Situation 1: In this case, exactly two of the coordinates of the source block B ($\alpha 1$, $\beta 1$, $\gamma 1$) have the same value as those of the corresponding coordinates in the destination block B ($\alpha 2$, $\beta 2$, $\gamma 2$). In this case, it may be readily verified, from the interconnection rules given above, that there exists a direct link between the source block and the destination block.

Situation 2: In this case, exactly one of the coordinates of the source block B ($\alpha 1$, $\beta 1$, $\gamma 1$) has the same value as that of the corresponding coordinate in the destination block B ($\alpha 2$, $\beta 2$, $\gamma 2$). In this case, it may be readily verified, from the interconnection rules given above that there exists an intermediate block B ($\alpha 3$, $\beta 3$, $\gamma 3$), such that there is a direct link between the source block B ($\alpha 1$, $\beta 1$, $\gamma 1$) and the intermediate block B ($\alpha 3$, $\beta 3$, $\gamma 3$) and a direct link between the intermediate block B ($\alpha 3$, $\beta 3$, $\gamma 3$) and the destination block. There are 3 possible choices for the values of ($\alpha 3$, $\beta 3$, $\gamma 3$) - ($\alpha 1$, $\beta 1$, $\gamma 2$), ($\alpha 1$, $\beta 2$, $\gamma 1$), ($\alpha 2$, $\beta 1$, $\gamma 1$).

Situation 3: In this case, none of the coordinates of the source block B ($\alpha 1, \beta 1, \gamma 1$) have the same value as that of the corresponding coordinate in the destination block B ($\alpha 2, \beta 2, \gamma 2$). In other words, $\alpha 1 \neq \alpha 2$, $\beta 1 \neq \beta 2$ and $\gamma 1 \neq \gamma 2$. In this case, there exist two intermediate blocks B ($\alpha 3, \beta 3, \gamma 3$) and B ($\alpha 4, \beta 4, \gamma 4$) such that there is a direct link between

- the source block B($\alpha 1, \beta 1, \gamma 1$) and the intermediate block B($\alpha 3, \beta 3, \gamma 3$),
- the block B($\alpha 3, \beta 3, \gamma 3$) and the block B($\alpha 4, \beta 4, \gamma 4$) and
- the block B ($\alpha 4, \beta 4, \gamma 4$) and the destination block B($\alpha 2, \beta 2, \gamma 2$).

There are a number of ways in which we may choose the intermediate blocks B ($\alpha 3, \beta 3, \gamma 3$) and B ($\alpha 4, \beta 4, \gamma 4$). For example, we could select B ($\alpha 2, \beta 1, \gamma 1$) and B ($\alpha 2, \beta 1, \gamma 2$) as intermediate nodes.

Since we use 6 coordinates to denote a processor, it is convenient to consider a 6-dimensional space where we have a point in that space, representing a processor whenever we specify all the 6 coordinates, (α, β, γ, x, y, z). A number of processors that share 5 of these components must lie on a line. In other words, we may visualize a line of processors by specifying any 5 of these 6 components. For example, in figure 3.5, by specifying (α, β, γ, 2, 2, *) we are specifying the processors (α, β, γ, 2, 2, 1), (α, β, γ, 2, 2, 2), (α, β, γ, 2, 2, 3) which are next to one another and forms a line of processors. Extending the idea, if we specify any 4 of these 6 components, we define a plane.

For example by specifying (α, β, γ, 2, *, *) we are specifying the following lines of processors:

- (α, β, γ, 2, 1, 1), (α, β, γ, 2, 1, 2), (α, β, γ, 2, 1, 3)
- (α, β, γ, 2, 2, 1), (α, β, γ, 2, 2, 2), (α, β, γ, 2, 2, 3),
- (α, β, γ, 2, 3, 1), (α, β, γ, 2, 3, 1), (α, β, γ, 2, 3, 1),

Theorem 1: There always exists a path of length $\leq 3n$ from any processor P (α1, β1, γ1, x1, y1, z1) to any other processor P (α2, β2, γ2, x2, y2, z2).

Proof:

If we divide the source block by three imaginary planes - (α1, β1, γ1, β2, *, *), (α1, β1, γ1, *, γ2, *) and (α1, β1, γ1, *, *, α2) as we show in figure 3.5, we get 8 octants in the source block which we will denote as SO1, SO2, SO3, SO4, SO5, SO6, SO7 and SO8. Similarly by dividing the destination block by three other imaginary planes (α2, β2, γ2, β1, *, *), (α2, β2, γ2, *, γ1, *) and (α2, β2, γ2, *, *, α1) we'll get 8 octants- DO1, DO2, DO3, DO4, DO5, DO6, DO7 and DO8 in the destination block.

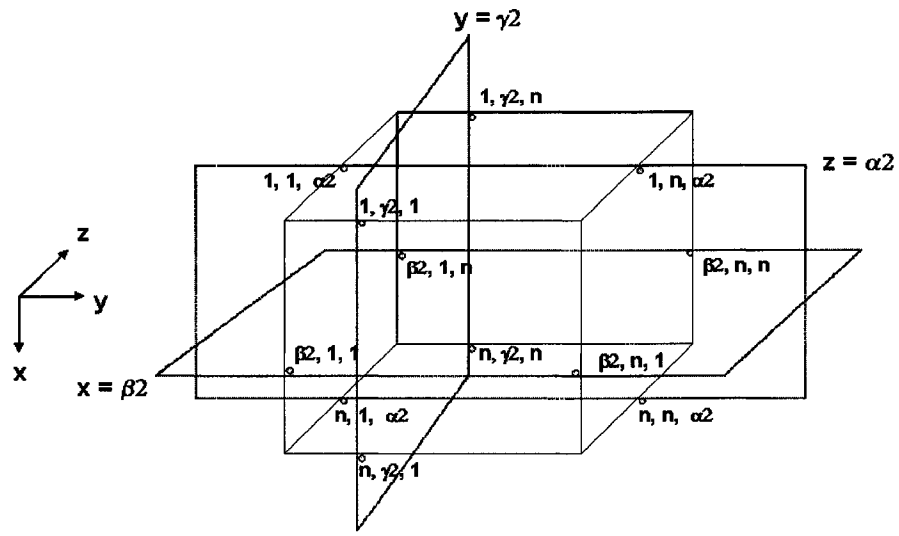

Figure 3.5: Three imaginary planes divide the source block into 8 octants

Since the source (destination) node **S** = P ($\alpha 1$, $\beta 1$, $\gamma 1$, x1, y1, z1) (respectively **D** = P ($\alpha 2$, $\beta 2$, $\gamma 2$, x2, y2, z2)) may be in any one of the 8 octants in the source (destination) block, we have to consider 64 possible octet pair combinations for the source destination pair (**S**, **D**). To illustrate our approach, we will only consider the case where the source (destination) node is in the octet SO1 (DO1). In other words, $1 \leq x1 \leq \beta 2$, $1 \leq y1 \leq \gamma 2$, $1 \leq z1 \leq \alpha 2$, $1 \leq x2 \leq \beta 1$, $1 \leq y2 \leq \gamma 1$, $1 \leq z2 \leq \alpha 1$. A possible path PT1 from the source node (which is in the block ($\alpha 1$, $\beta 1$, $\gamma 1$)) to the destination node (in the block ($\alpha 2$, $\beta 2$, $\gamma 2$)) using the intermediate blocks ($\alpha 2$, $\beta 1$, $\gamma 1$) and ($\alpha 2$, $\beta 2$, $\gamma 1$) may be formulated as follows-

Path PT1:

P ($\alpha 1$, $\beta 1$, $\gamma 1$, x1, y1, z1) $\rightarrow$ ... $\rightarrow$ P ($\alpha 1$, $\beta 1$, $\gamma 1$, 1, y1, $\alpha 2$) $\rightarrow$ P ($\alpha 2$, $\beta 1$, $\gamma 1$, *n*, y1, $\alpha 1$) $\rightarrow$ ... $\rightarrow$ P ($\alpha 2$, $\beta 1$, $\gamma 1$, $\beta 2$, 1, $\alpha 1$) $\rightarrow$ P ($\alpha 2$, $\beta 2$, $\gamma 1$, $\beta 1$, *n*, $\alpha 1$) $\rightarrow$ ... $\rightarrow$ P ($\alpha 2$, $\beta 2$, $\gamma 1$, $\beta 1$, $\gamma 2$, *n*) $\rightarrow$ P($\alpha 2$, $\beta 2$, $\gamma 2$, $\beta 1$, $\gamma 1$, 1) $\rightarrow$ ... $\rightarrow$ P ($\alpha 2$, $\beta 2$, $\gamma 2$, x2,

y2, z2).

The length of this path is $\mathbf{L}_{PT1}$ where

$\mathbf{L}_{PT1}$ = (x1 - 1) + (α2 – z1) + 1 + (*n*- β2) + (y1 – 1) + 1 + (*n* - γ2) + (*n* - α1) + 1 + (β1- x2) + (γ1 – y2) + (z2 - 1)

= x1-1+α2–z1+1+*n*-β2+ y1– 1 + 1 + *n* - γ2 + *n* - α1 + 1 + β1- x2 + γ1– y2 + z2 -1

= 3*n* + x1 + y1 - z1 - α1 + β1 + γ1 - x2 - y2 + z2 + α2 - β2 - γ2.

In a similar way, a possible path PT2 from the source node to the destination node using the intermediate blocks (α1, β1, γ2) and (α1, β2, γ2) may be formulated as follows-

Path PT2:

P (α1, β1, γ1, x1, y1, z1) → ... → P (α1, β1, γ1, x1, γ2, 1) → P (α1, β1, γ2, x1, γ1, *n*) → ... → P (α1, β1, γ2, β2, *n*, *n*) → P (α1, β2, γ2, β1, 1, *n*) → ... → P (α1, β2, γ2, *n*, 1, α2) → P (α2, β2, γ2, 1, 1, α1) → ... → P (α2, β2, γ2, x2, y2, z2).

The length of this path is $\mathbf{L}_{PT2}$ where

$\mathbf{L}_{PT2}$ = (γ2 - y1) + (z1 - 1) + 1 + (β2 - x1) + (*n* - γ1) + 1 + (*n* - β1) + (*n* - α2) + 1 + (x2 - 1) + (y2 – 1) + (α1- z2)

= γ2-y1+ z1-1+1+β2 - x1 + *n* - γ1 + 1 + *n* - β1 + *n* - α2 + 1 + x2 - 1 + y2 – 1+ α1-z2

= 3*n* – x1 – y1 + z1 + α1 - β1 - γ1 + x2 + y2 – z2 - α2 + β2 + γ2.

It may be readily verified that the sum of these two path lengths are $\mathbf{L}_{PT1} + \mathbf{L}_{PT2} = 6n$. Therefore the smaller of these two paths must be $3n$ or less.

For other 63 possible cases of source and destination processor locations in various octants have been checked in the similar way.

Next we show that there exists at least one source and destination pair in the network whose minimum distance is $3n$.We have to consider two situations - n is even and n is odd

1. If n is even, let us consider the source processor P(1,1,1, 1, 1,1,) and the destination processor $P(\frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1)$.

2. If n is odd, we consider the source P (1, 1, 1, 1, 1, 1) and the destination $P(\frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2})$.

Situation 1: P (1,1,1, 1, 1,1,) to $P(\frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1)$, n is even

$P(1, 1, 1, 1, 1, 1) \rightarrow P(1, 1, 1, 1, 1, \frac{n}{2}+1) \rightarrow P(\frac{n}{2}+1, 1, 1, n, 1, 1) \rightarrow P(\frac{n}{2}+1, 1, 1, \frac{n}{2}+1, 1, 1) \rightarrow P(\frac{n}{2}+1, \frac{n}{2}+1, 1, 1, n, 1) \rightarrow P(\frac{n}{2}+1, \frac{n}{2}+1, 1, 1, \frac{n}{2}+1, 1) \rightarrow P(\frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1, 1, 1, n) \rightarrow P(\frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1, \frac{n}{2}+1)$

The cost of this path is-

$$(\frac{n}{2}+1-1)+1+(n-\frac{n}{2}-1)+1+(n-\frac{n}{2}-1)+1+(\frac{n}{2}+1-1)+(\frac{n}{2}+1-1)+(n-\frac{n}{2}-1)$$

$$=\frac{n+2+2n-n-2+2+2n-n-2+2+n+n+2n-n-2}{2}$$

$$=\frac{9n-3n}{2}$$

$$=3n$$

Situation 2: P (1, 1, 1, 1, 1, 1) to P ($\frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2}$) , n is odd

P (1, 1, 1, 1, 1, 1) $\rightarrow$ P (1, 1, 1, 1, 1, $\frac{n+1}{2}$) $\rightarrow$ P ($\frac{n+1}{2}$, 1, 1, n, 1, 1) $\rightarrow$ P ($\frac{n+1}{2}$, 1, 1, $\frac{n+1}{2}$, 1 , 1) $\rightarrow$ P ($\frac{n+1}{2}, \frac{n+1}{2}$, 1 , n, 1) $\rightarrow$ P ($\frac{n+1}{2}, \frac{n+1}{2}$, 1, 1 , $\frac{n+1}{2}$, 1) $\rightarrow$ P ($\frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2}$, 1, 1, n) $\rightarrow$ P ($\frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2}, \frac{n+1}{2}$)

The cost of this path is:

$$(\frac{n+1}{2} - 1) + 1 + (n - \frac{n+1}{2}) + 1 + (n - \frac{n+1}{2}) + 1 + (\frac{n+1}{2} - 1) + (\frac{n+1}{2} - 1) + (n - \frac{n+1}{2})$$

$$= \frac{n+1-2+2+2n-n-1+2+2n-n-1+2+n+1-2+n+1-2+2n-n-1}{2}$$

$$= \frac{9n-3n+9-9}{2}$$

$$= 3n$$

The following path shows an example from corner to corner- the source processor is P (1, 1, 1, 1, 1, 1) and the destination processor is P (n, n, n, n, n, n) and

Path: P (1, 1, 1, 1, 1, 1) $\rightarrow$ P (1, 1, 1, 1, 1, n) $\rightarrow$ P (n, 1, 1, n, 1, 1) $\rightarrow$ P (n, n, 1, 1, n, 1) $\rightarrow$ P (n, n, n, 1, 1, n) $\rightarrow$ P (n, n, n, n, n, n)

The length of this path is $(n-1) + 1 + 1 + 1 + (n-1) + (n-1) = 3n$

The diameter of the 3D MM is only O ($N^{1/6}$) in contrast to O ($N^{1/3}$) on a 3-dimensional torus with the same node degree of 6. We note that the Multi-Mesh has a diameter of O ($N^{1/4}$) with a node degree of 4 that was shown to be attractive with respect to other topologies [DDS99], [HwBr83], [Le92].

The node degree and the diameters of the Hypercube, the Multi-Mesh and the 3D MM containing N processors given in table 3.1.

Table 3.1: Diameter of Hypercube, Multi-Mesh and 3D MM

Hypercube		Multi-Mesh		3D MM	
Node degree	Diameter	Node degree	Diameter	Node degree	Diameter
$\log_2 N$	$\log_2 N$	4	$N^{1/4}$	6	$N^{1/6}$

As an example, Table 3.2 shows a comparison between the diameter of a hypercube, Multi-Mesh and 3D MM network for different total number of nodes (N).

Table 3.2: An example of Diameter of Hypercube, Multi-Mesh and 3D MM

# of nodes	Hypercube		Multi-Mesh		3D MM	
	Node degree	Diameter	Node degree	Diameter	Node degree	Diameter
64	6	6	4	6	6	6
4096	12	12	4	16	6	12
256K	18	18	4	44	6	24
16M	24	24	4	126	6	48

Thus, for $N = 4096$, the diameter of both the 3D MM network and the binary hypercube is equal to 12, but the node degree of the corresponding hypercube is 12, while that of the 3D MM network is only 6. In other words the diameter for 3D MM networks with 4096 processors is less and the node degree is constant.

3.2.2 Connectivity of Multi-Mesh network

According to D. Sima, T.Fountain, P.Kacsuk [Be73], [SFK97], the connectivity of a graph is defined as the minimum number of arcs of a connected graph that have to be removed in order that the resulting sub-graph consists of two disconnected sub-graphs. It is also well known that if the connectivity of a graph is C, we can always find C node disjoint paths between any pair of nodes [SBS01].

An interconnection network with a higher connectivity is preferable since higher connectivity implies better fault tolerance and higher capability for load balancing.

The Multi-Mesh (MM) network that we described in chapter 2 is a *regular graph* [Be73] where the node degree of each processor in the network is four. As a result, the upper bound of connectivity of any MM is four. In this section we will prove that the connectivity of MM is exactly four.

As we described in Chapter 2, the two-dimensional mesh is the basic building block in a Multi-Mesh network. In [DDS99], based on the position of a processor within a block, the processor was classified into the following categories:

1) internal processor
2) boundary processor
3) corner processor.

Within a block, an internal processor has exactly four neighbor processor (connected by intra-block links), a boundary processor has three neighbors and a corner processor has two neighbors. D. De, D and B.P. Sinha [DDS99] shows how the inter-block links of a MM network ensure that each processor in a MM network has exactly four links.

Theorem 2:

The connectivity of a Multi-Mesh network is 4.

Proof

In order to prove this, we have to show that, regardless of the position of the source and the destination, we can always find 4 edge-disjoint paths **ED1**, **ED2**, **ED3** and **ED4**. The source and the destination may be in the same block or in different blocks. We will discuss only the case where they are in different blocks since that is the more challenging task.

We need to consider 9 possible combinations of source and destination processor categories. We will consider the following two cases -

- **Case 1:** The source and the destination are both internal processors,
- **Case 2:** The source is a boundary processor and the destination is an internal processor.

The remaining 7 can be handled in the similar way.

Case 1: The source and the destination are both internal processors.

If the source and destination are both internal, then the following conditions are hold:

i) $1 < x1 < \beta 2$ and $1 < y1 < \alpha 2$

ii) $1 < x2 < \beta 1$ and $1 < y2 < \alpha 1$

Since both the source node and the destination node are internal processors, they both have 4 neighbors. We now show how we may create four edge disjoint paths **ED1**, **ED2**, **ED3** and **ED4** from the source to the destination node.

For each path, we

a) first give the path at the block level where we only specify the blocks used in the path,

b) then give a short description of the path,

c) finally give a detailed description of the path used.

In giving a short description of a path, we have used the notation $X \rightarrow^* Y$ to denote that we have used a number of intra-block edges to go from processor X to processor Y.

Path ED1:

a) At the block level the path is as follows:

B ($\alpha 1$, $\beta 1$) $\rightarrow$ B ($\alpha 2$, $\beta 1$) $\rightarrow$ B ($\alpha 2$, $\beta 2$).

b) A short description of the path is as follows:

P ($\alpha 1$, $\beta 1$, x1, y1) $\rightarrow^*$ P ($\alpha 1$, $\beta 1$, 1, $\alpha 2$) $\rightarrow$ P ($\alpha 2$, $\beta 1$, n, $\alpha 1$) $\rightarrow^*$ P ($\alpha 2$, $\beta 1$, $\beta 2$, n) $\rightarrow$ P ($\alpha 2$, $\beta 2$, $\beta 1$, 1) $\rightarrow^*$ P ($\alpha 2$, $\beta 2$, x2, y2).

c) A detailed description of the path used is as follows:

P ($\alpha 1$, $\beta 1$, x1, y1) $\rightarrow$ P ($\alpha 1$, $\beta 1$, x1-1, y1) $\rightarrow$ ... $\rightarrow$ P ($\alpha 1$, $\beta 1$, 1, y1) $\rightarrow$
P ($\alpha 1$, $\beta 1$, 1, y1 + 1) $\rightarrow$ ... $\rightarrow$ P ($\alpha 1$, $\beta 1$, 1, $\alpha 2$) $\rightarrow$ P ($\alpha 2$, $\beta 1$, n, $\alpha 1$) $\rightarrow$
P ($\alpha 2$, $\beta 1$, n, $\alpha 1$ + 1) $\rightarrow$ ... $\rightarrow$ P ($\alpha 2$, $\beta 1$, n, n) $\rightarrow$ P ($\alpha 2$, $\beta 1$, n - 1, n) $\rightarrow$ ... $\rightarrow$
P ($\alpha 2$, $\beta 1$, $\beta 2$, n) $\rightarrow$ P ($\alpha 2$, $\beta 2$, $\beta 1$, 1) $\rightarrow$ P ($\alpha 2$, $\beta 2$, $\beta 1$ – 1, 1) $\rightarrow$ ... $\rightarrow$
P ($\alpha 2$, $\beta 2$, x2, 1) $\rightarrow$ P ($\alpha 2$, $\beta 2$, x2, 1+1) $\rightarrow$ ... $\rightarrow$ P ($\alpha 2$, $\beta 2$, x2, y2).

Path ED2:

a) At the block level the path is as follows:

B ($\alpha 1$, $\beta 1$) $\rightarrow$ B ($\alpha 2$, $\beta 1$) $\rightarrow$ B ($\alpha 2$, $\beta 2$)

b) A short description of the path is as follows:

P ($\alpha 1$, $\beta 1$, x1, y1) $\rightarrow^*$ P ($\alpha 1$, $\beta 1$, n, $\alpha 2$) $\rightarrow$ P ($\alpha 2$, $\beta 1$, 1, $\alpha 1$) $\rightarrow^*$ P ($\alpha 2$, $\beta 1$, $\beta 2$,1)

$\rightarrow$ P (α2, β2, β1, n)* $\rightarrow$ P (α2, β2, x2, y2).

c) A detailed description of the path used is as follows:

P (α1, β1, x1, y1) $\rightarrow$ P (α1, β1, x1+ 1, y1) $\rightarrow$ ... $\rightarrow$ P (α1, β1, n, y1) $\rightarrow$

P (α1, β1, n, y1 + 1) $\rightarrow$ ... $\rightarrow$P (α1, β1, n, α2) $\rightarrow$ P (α2, β1, 1, α1) $\rightarrow$

P (α2, β1, 1, α1$-$1) $\rightarrow$ ... $\rightarrow$ P (α2, β1, 1, 1) $\rightarrow$ P (α2, β1, 1+ 1, 1) $\rightarrow$ ... $\rightarrow$

P (α2, β1, β2, 1) $\rightarrow$ P (α2, β2, β1, n) $\rightarrow$ P (α2, β2, β1 $-$ 1, n) $\rightarrow$ ... $\rightarrow$

P (α2, β2, x2, n) $\rightarrow$ P (α2, β2, x2, n -1) $\rightarrow$ ... $\rightarrow$ P (α2, β2, x2, y2).

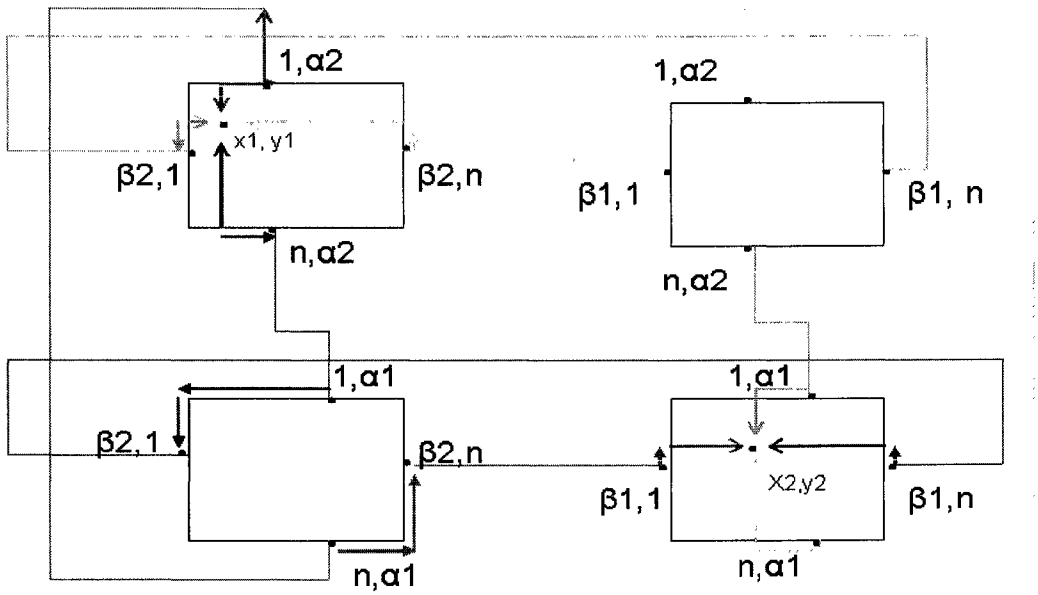

Figure 3.6: Possible four disjoint paths from source to destination (Case 1)

Path ED3:

a) At the block level the path is as follows:

B (α1, β1) $\rightarrow$ B (α1, β2) $\rightarrow$ B (α2, β2)

b) A short description of the path is as follows:

P (α1, β1, x1, y1) $\rightarrow^*$ P (α1, β1, β2, 1) $\rightarrow$ P (α1, β2, β1, n) $\rightarrow^*$ P (α1, β2, n, α2)

$\rightarrow$ P (α2, β2, 1, α1) $\rightarrow^*$ P (α2, β2, x2, y2).

c) A detailed description of the path used is as follows:

P (α1, β1, x1, y1) $\rightarrow$ P (α1, β1, x1, y1- 1) $\rightarrow$ ... $\rightarrow$ P (α1, β1, x1, 1) $\rightarrow$
P (α1, β1, x1+1, 1) $\rightarrow$ ... $\rightarrow$ P (α1, β1, β2, 1) $\rightarrow$ P (α1, β2, β1, *n*) $\rightarrow$
P (α1, β2, β1+1, *n*) $\rightarrow$ ... $\rightarrow$ P (α1, β2, *n*, *n*) $\rightarrow$ P (α1, β2, *n*, *n*-1) $\rightarrow$ ... $\rightarrow$
P (α1, β2, *n*, α2) $\rightarrow$ P (α2, β2, 1, α1) $\rightarrow$ P (α2, β2, 1, α1−1) $\rightarrow$ ... $\rightarrow$
P (α2, β2, 1, y2) $\rightarrow$ P (α2, β2, 1 + 1, y2) $\rightarrow$ ... $\rightarrow$ P (α2, β2, x2, y2).

Path ED4:

a) At the block level the path is as follows:

B (α1, β1) $\rightarrow$ B (α1, β2) $\rightarrow$ B(α2, β2)

b) A short description of the path is as follows:

P (α1, β1, x1, y1) $\rightarrow^*$ P (α1, β1, β2, *n*) $\rightarrow$ P (α1, β2, β1, 1) $\rightarrow$ *P (α1, β2, 1, α2)
$\rightarrow$ P (α2, β2, *n*, α1) $\rightarrow^*$ P (α2, β2, x2, y2).

c) A detailed description of the path used is as follows:

P (α1, β1, x1, y1) $\rightarrow$ P (α1, β1, x1, y1+ 1) $\rightarrow$ ... $\rightarrow$ P (α1, β1, x1, *n*) $\rightarrow$
P (α1, β1, x1+1, *n*) $\rightarrow$ ... $\rightarrow$ P (α1, β1, β2, *n*) $\rightarrow$ P (α1, β2, β1, 1) $\rightarrow$
P (α1, β2, β1−1, 1) $\rightarrow$ ... $\rightarrow$ P (α1, β2, 1, 1) $\rightarrow$ P (α1, β2, 1, 1+1) $\rightarrow$ ... $\rightarrow$
P (α1, β2, 1, α2) $\rightarrow$ P (α2, β2, *n*, α1) $\rightarrow$ P (α2, β2, *n*, α1−1) $\rightarrow$ ... $\rightarrow$
P (α2, β2, *n*, y2) $\rightarrow$ P (α2, β2, *n*-1, y2) $\rightarrow$ ... $\rightarrow$ P (α2, β2, x2, y2).

Case 2: Source processor is a boundary processor and destination processor is an internal processor. In this case there are three links to other neighboring processor and the other link is with a processor situated on the other block.

We will consider the situation where the following conditions hold:

i) $x1 = 1$ and $x1 < \beta 2$ and $1 < y1 < \alpha 2$

ii) $1 < x2 < \beta 1$ and $1 < y2 < \alpha 1$

Remaining situations can be solved in the similar way. We now show how we may create four edge disjoint paths **ED1**, **ED2, ED3** and **ED4** from the source to the destination node.

Path ED1:

a) At the block level the path is as follows:

$B(\alpha 1, \beta 1) \rightarrow B(\alpha 2, \beta 1) \rightarrow B(\alpha 2, \beta 2)$

b) A short description of the path is as follows:

$P(\alpha 1, \beta 1, 1, y1) \rightarrow^{*} P(\alpha 1, \beta 1, 1, \alpha 2) \rightarrow P(\alpha 2, \beta 1, n, \alpha 1) \rightarrow^{*} P(\alpha 2, \beta 1, \beta 2, n)$
$\rightarrow P(\alpha 2, \beta 2, \beta 1, 1) \rightarrow^{*} P(\alpha 2, \beta 2, x2, y2).$

c) A detailed description of the path used is as follows:

$P(\alpha 1, \beta 1, 1, y1) \rightarrow P(\alpha 1, \beta 1, 1, y1 + 1) \rightarrow \ldots \rightarrow P(\alpha 1, \beta 1, 1, \alpha 2) \rightarrow$
$P(\alpha 2, \beta 1, n, \alpha 1) \rightarrow P(\alpha 2, \beta 1, n, \alpha 1 + 1) \rightarrow \ldots \rightarrow P(\alpha 2, \beta 1, n, n) \rightarrow$
$P(\alpha 2, \beta 1, n - 1, n) \rightarrow \ldots \rightarrow P(\alpha 2, \beta 1, \beta 2, n) \rightarrow P(\alpha 2, \beta 2, \beta 1, 1) \rightarrow$
$P(\alpha 2, \beta 2, \beta 1 - 1, 1) \rightarrow \ldots \rightarrow P(\alpha 2, \beta 2, x2, 1) \rightarrow P(\alpha 2, \beta 2, x2, 1+1) \rightarrow \ldots \rightarrow$
$P(\alpha 2, \beta 2, x2, y2).$

Path ED2:

a) At the block level the path is as follows:

$B(\alpha 1, \beta 1) \rightarrow B(\alpha 2, \beta 1) \rightarrow B(\alpha 2, \beta 2)$

b) A short description of the path is as follows:

$P(\alpha 1, \beta 1, 1, y1) \rightarrow^{*} P(\alpha 1, \beta 1, n, \alpha 2) \rightarrow P(\alpha 2, \beta 1, 1, \alpha 1) \rightarrow^{*} P(\alpha 2, \beta 1, \beta 2, 1)$

$\rightarrow$ P (α2, β2, β1, n) $\rightarrow^*$ P (α2, β2, x2, y2).

c) A detailed description of the path used is as follows:

P (α1, β1, 1, y1) $\rightarrow$ P (α1, β1, 1+ 1, y1) $\rightarrow$ ... $\rightarrow$ P (α1, β1, n, y1) $\rightarrow$

P (α1, β1, n, y1 + 1) $\rightarrow$ ... $\rightarrow$ P (α1, β1, n, α2) $\rightarrow$ P (α2, β1, 1, α1) $\rightarrow$

P (α2, β1, 1, α1–1) $\rightarrow$ ... $\rightarrow$P (α2, β1, 1, 1) $\rightarrow$ P (α2, β1, 1+ 1, 1) $\rightarrow$ ... $\rightarrow$

P (α2, β1, β2, 1) $\rightarrow$ P (α2, β2, β1, n) $\rightarrow$ P (α2, β2, β1 – 1, n) $\rightarrow$ ... $\rightarrow$

P (α2, β2, x2, n) $\rightarrow$ P (α2, β2, x2, n -1) $\rightarrow$ ... $\rightarrow$ P (α2, β2, x2, y2).

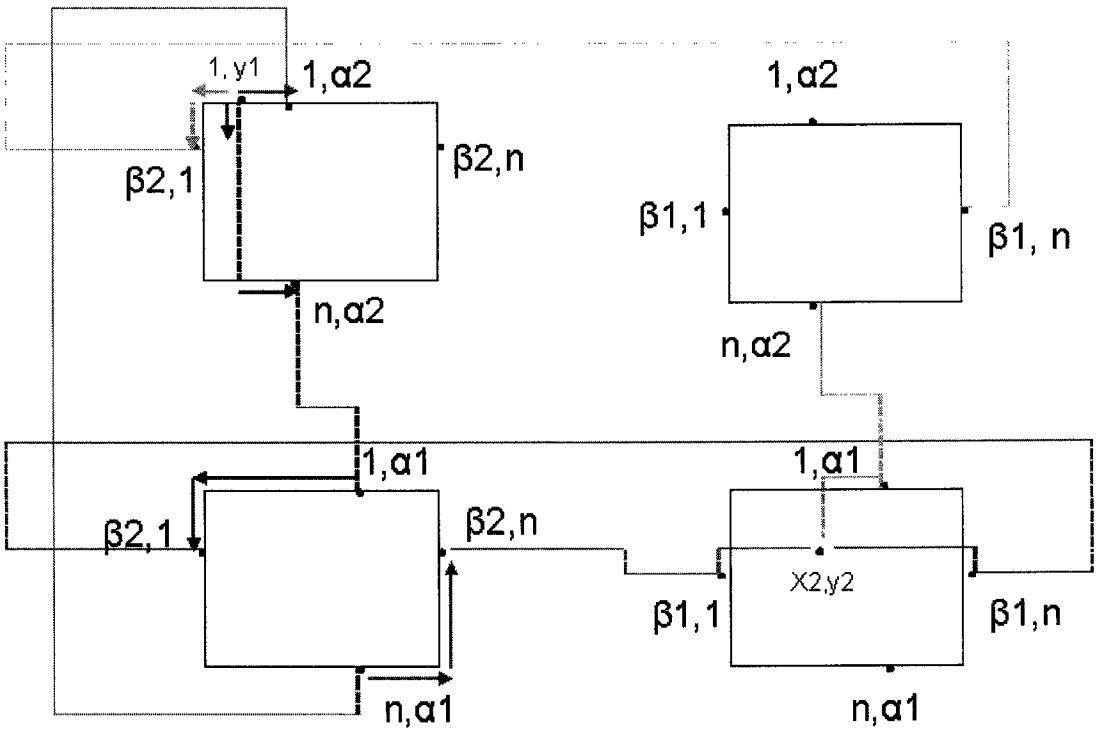

Figure 3.7: Possible four disjoint paths from source to destination (Case 2)

Path ED3:

a) At the block level the path is as follows:

B(α1, β1) $\rightarrow$ B(α1, β2) $\rightarrow$ B(α2, β2)

b) A short description of the path is as follows:

$P(\alpha1, \beta1, 1, y1) \rightarrow^* P(\alpha1, \beta1, \beta2, 1) \rightarrow P(\alpha1, \beta2, \beta1, n) \rightarrow^* P(\alpha1, \beta2, n, \alpha2)$
$\rightarrow P(\alpha2, \beta2, 1, \alpha1) \rightarrow^* P(\alpha2, \beta2, x2, y2)$.

c) A detailed description of the path used is as follows:

$P(\alpha1, \beta1, 1, y1) \rightarrow P(\alpha1, \beta1, 1, y1- 1) \rightarrow ... \rightarrow P(\alpha1, \beta1, 1, 1) \rightarrow$
$P(\alpha1, \beta1, 1+1, 1) \rightarrow ... \rightarrow P(\alpha1, \beta1, \beta2, 1) \rightarrow P(\alpha1, \beta2, \beta1, n) \rightarrow$
$P(\alpha1, \beta2, \beta1+1, n) \rightarrow ... \rightarrow P(\alpha1, \beta2, n, n) \rightarrow P(\alpha1, \beta2, n, n-1) \rightarrow ... \rightarrow$
$P(\alpha1, \beta2, n, \alpha2) \rightarrow P(\alpha2, \beta2, 1, \alpha1) \rightarrow P(\alpha2, \beta2, 1, \alpha1-1) \rightarrow ... \rightarrow$
$P(\alpha2, \beta2, 1, y2) \rightarrow P(\alpha2, \beta2, 1 + 1, y2) \rightarrow ... \rightarrow P(\alpha2, \beta2, x2, y2)$.

Path ED4:

There is an inter-block link connecting the processors $P(\alpha1, \beta1, 1, y1)$ and $P(y1, \beta1, n, \alpha1)$. If $y1 = \alpha1$ we have a self loop. Otherwise we will reach another block $B(y1, \beta1)$. If it is on another block then obviously we'll get another distinct path.

So there are two situations-

i) $y1 \neq \alpha1$

ii) $y1 = \alpha1$

For situation 1, if $y1 \neq \alpha1$, then in order to get PT4 an edge-disjoint path we will take the following paths-

a) At the block level the path is as follows:

$B(\alpha1, \beta1) \rightarrow B(y1, \beta1) \rightarrow B(y1, \beta2) \rightarrow B(\alpha2, \beta2)$

b) A short description of the path is as follows:

$P(\alpha1, \beta1, 1, y1) \rightarrow P(y1, \beta1, n, \alpha1) \rightarrow^* P(y1, \beta1, \beta2, 1) \rightarrow P(y1, \beta2, \beta1, n) \rightarrow^*$
$P(y1, \beta2, 1, \alpha2) \rightarrow P(\alpha2, \beta2, n, y1) \rightarrow^* P(\alpha2, \beta2, x2, y2)$.

c) A detailed description of the path used is as follows:

$P(\alpha1, \beta1, 1, y1) \rightarrow P(y1, \beta1, n, \alpha1) \rightarrow P(y1, \beta1, n\text{-}1, \alpha1) \rightarrow ... \rightarrow$

$P(y1, \beta1, \beta2, \alpha1) \rightarrow P(y1, \beta1, \beta2, \alpha1\text{-}1) \rightarrow ... \rightarrow P(y1, \beta1, \beta2, 1) \rightarrow$

$P(y1, \beta2, \beta1, n) \rightarrow P(y1, \beta2, \beta1{-}1, 1) \rightarrow ... \rightarrow P(y1, \beta2, 1, n) \rightarrow P(y1, \beta2, 1, n - 1)$

$\rightarrow ... \rightarrow P(y1, \beta2, 1, \alpha2) \rightarrow P(\alpha2, \beta2, n, y1) \rightarrow P(\alpha2, \beta2, n\text{-}1, y1) \rightarrow ... \rightarrow$

$P(\alpha2, \beta2, x2, y1) \rightarrow P(\alpha2, \beta2, x2, y1\text{-}1) \rightarrow ... \rightarrow P(\alpha2, \beta2, x2, y2).$

For situation 2, if $y1 = \alpha1$, then in order to get PT4 an edge-disjoint path we will take the following paths-

a) At the block level the path is as follows:

$B(\alpha1, \beta1) \rightarrow B(\alpha1{-}1, \beta1) \rightarrow B(\alpha1{-}1, \beta2) \rightarrow B(\alpha2, \beta2)$

b) A short description of the path is as follows:

$P(\alpha1, \beta1, 1, \alpha1) \rightarrow P(\alpha1, \beta1, n, \alpha1) \rightarrow P(\alpha1, \beta1, n, \alpha1\text{-}1) \rightarrow P(\alpha1\text{-}1, \beta1, 1, \alpha1)$

$\rightarrow^{*} P(\alpha1\text{-}1, \beta1, \beta2, 1) \rightarrow P(\alpha1{-}1, \beta2, \beta1, n) \rightarrow^{*} P(\alpha1{-}1, \beta2, n, \alpha2)$

$\rightarrow P(\alpha2, \beta2, 1, \alpha1{-}1) \rightarrow^{*} P(\alpha2, \beta2, x2, y2).$

c) A detailed description of the path used is as follows:

$P(\alpha1, \beta1, 1, \alpha1) \rightarrow P(\alpha1, \beta1, n, \alpha1) \rightarrow P(\alpha1, \beta1, n, \alpha1\text{-}1) \rightarrow P(\alpha1\text{-}1, \beta1, 1, \alpha1)$

$\rightarrow P(\alpha1\text{-}1, \beta1, 1, \alpha1\text{-}1) \rightarrow ... \rightarrow P(\alpha1\text{-}1, \beta1, 1, 1) \rightarrow P(\alpha1\text{-}1, \beta1, 1{+}1, 1) \rightarrow ... \rightarrow P$

$(\alpha1\text{-}1, \beta1, \beta2, 1) \rightarrow P(\alpha1{-}1, \beta2, \beta1, n) \rightarrow P(\alpha1{-}1, \beta2, \beta1\text{-}1, n) \rightarrow ... \rightarrow P(\alpha1{-}1,$

$\beta2, 1, n) \rightarrow P(\alpha1{-}1, \beta2, 1, n\text{-}1) \rightarrow ... \rightarrow P(\alpha1{-}1, \beta2, 1, \alpha2) \rightarrow P(\alpha2, \beta2, n,$

$\alpha1{-}1) \rightarrow P(\alpha2, \beta2, n, \alpha1{-}1{+}1) \rightarrow ... \rightarrow P(\alpha2, \beta2, n, y2) \rightarrow P(\alpha2, \beta2, n\text{-}1, y2)$

$\rightarrow ... \rightarrow P(\alpha2, \beta2, x2, y2).$

It may be readily verified that these paths are edge disjoint.

3.2.3 Connectivity of three dimensional Multi-Mesh (3D MM) network

Our proposed network is a three dimensional Multi-Mesh (3D MM) where we defined the interconnection rules in such a way so that we will get a regular graph where the node degree of each processor in the network is six. As a result, the upper bound of connectivity of 3D MM is six. In this section we will prove that the connectivity of 3D MM is exactly six.

As we discussed, the basic building block in a 3D MM is a three dimensional mesh. Depending on the position of a processor within a block, the processor was classified into the following categories (discussed in the section 3.2.2):

1) internal processor,
2) face-centered processors
3) boundary edge processor,
4) corner processor and

Within a block, an internal processor has exactly six links to other neighbors, a face-centered processor has five neighbors, a boundary edge processor has four links to its neighbors, a corner processor has three links to its neighbors. By defining the inter-block links we ensure that every processor has exactly six links to other processors.

Theorem 3:

The connectivity of a three dimensional multi-mesh (3D MM) network is 6.

Proof

In order to prove this, we have to show that, regardless of the position of the source and the destination, we can always find 6 edge-disjoint paths **EDPT1**, **EDPT2**, **EDPT3**,

EDPT4, **EDPT5** and **EDPT6**. The source and the destination may be in the same block or in different blocks. We will discuss only the case where they are in different blocks since that is the more challenging task.

We need to consider 16 possible combinations of source and destination processor categories. We will only show the following case – where source and destination both are internal processors.

We will consider the situation where the following conditions hold:

i) $1 < x1 < \beta 2$, $1 < y1 < \gamma 2$ and $1 < z1 < \alpha 2$

ii) $1 < x2 < \beta 1$, $1 < \gamma 2 < \alpha 1$ and $1 < z2 < \alpha 1$

Since both the source node and the destination node are internal processors, they both have 6 neighbors. We now show how we may create six edge disjoint paths **EDPT1**, **EDPT2**, **EDPT3**, **EDPT4**, **EDPT5** and **EDPT6** from the source to the destination node. For each path, we

a) first give the path at the block level where we only specify the blocks used in the path,
b) then give a short description of the path,
c) finally give a detailed description of the path used.

In giving a short description of a path, we have used the notation $X \rightarrow^{*} Y$ to denote that we have used a number of intra-block edges to go from processor X to processor Y.

Path EDPT1:

a) At the block level the path is as follows:

B $(\alpha 1, \beta 1, \gamma 1) \rightarrow$ B $(\alpha 2, \beta 1, \gamma 1) \rightarrow$ B $(\alpha 2, \beta 2, \gamma 1) \rightarrow$ B $(\alpha 2, \beta 2, \gamma 2)$

b) A short description of the path is as follows:

$P(\alpha1, \beta1, \gamma1, x1, y1, z1) \rightarrow^* P(\alpha1, \beta1, \gamma1, 1, y1, \alpha2) \rightarrow P(\alpha2, \beta1, \gamma1, n, y1, \alpha1)$ $\rightarrow^* P(\alpha2, \beta1, \gamma1, \beta2, 1, \alpha1) \rightarrow P(\alpha2, \beta2, \gamma1, \beta1, n, \alpha1) \rightarrow^* P(\alpha2, \beta2, \gamma1, \beta1, \gamma2, n) \rightarrow P(\alpha2, \beta2, \gamma2, \beta1, \gamma1, 1) \rightarrow^* P(\alpha2, \beta2, \gamma2, x2, y2, z2)$.

c) A detailed description of the path used is as follows:

$P(\alpha1, \beta1, \gamma1, x1, y1, z1) \rightarrow P(\alpha1, \beta1, \gamma1, x1\text{-}1, y1, z1) \rightarrow \ldots \rightarrow P(\alpha1, \beta1, \gamma1, 1, y1, z1) \rightarrow P(\alpha1, \beta1, \gamma1, 1, y1, z1+1) \rightarrow \ldots \rightarrow P(\alpha1, \beta1, \gamma1, 1, y1, \alpha2) \rightarrow P(\alpha2, \beta1, \gamma1, n, y1, \alpha1) \rightarrow P(\alpha2, \beta1, \gamma1, n, y1\text{-}1, \alpha1) \rightarrow \ldots \rightarrow P(\alpha2, \beta1, \gamma1, n, 1, \alpha1) \rightarrow P(\alpha2, \beta1, \gamma1, n\text{-}1, 1, \alpha1) \rightarrow \ldots \rightarrow P(\alpha2, \beta1, \gamma1, \beta2, 1, \alpha1) \rightarrow P(\alpha2, \beta2, \gamma1, \beta1, n, \alpha1) \rightarrow P(\alpha2, \beta2, \gamma1, \beta1, n, \alpha1+1) \rightarrow \ldots \rightarrow P(\alpha2, \beta2, \gamma1, \beta1, n, n) \rightarrow P(\alpha2, \beta2, \gamma1, \beta1, n - 1, n) \rightarrow \ldots \rightarrow P(\alpha2, \beta2, \gamma1, \beta1, \gamma2, n) \rightarrow P(\alpha2, \beta2, \gamma2, \beta1, \gamma1, 1) \rightarrow P(\alpha2, \beta2, \gamma2, \beta1 - 1, \gamma1, 1) \rightarrow \ldots \rightarrow P(\alpha2, \beta2, \gamma2, x2, \gamma1, 1) \rightarrow P(\alpha2, \beta2, \gamma2, x2, \gamma1 - 1, 1) \rightarrow \ldots \rightarrow P(\alpha2, \beta2, \gamma2, x2, y2, 1) \rightarrow P(\alpha2, \beta2, \gamma2, x2, y2, 1 + 1) \rightarrow \ldots \rightarrow P(\alpha2, \beta2, \gamma2, x2, y2, z2)$.

Path EDPT2:

a) At the block level the path is as follows:

$B(\alpha1, \beta1, \gamma1) \rightarrow B(\alpha2, \beta1, \gamma1) \rightarrow B(\alpha2, \beta1, \gamma2) \rightarrow B(\alpha2, \beta2, \gamma2)$

b) A short description of the path is as follows:

$P(\alpha1, \beta1, \gamma1, x1, y1, z1) \rightarrow^* P(\alpha1, \beta1, \gamma1, n, y1, \alpha2) \rightarrow P(\alpha2, \beta1, \gamma1, 1, y1, \alpha1) \rightarrow^* P(\alpha2, \beta1, \gamma1, 1, \gamma2, n) \rightarrow P(\alpha2, \beta1, \gamma2, 1, \gamma1, 1) \rightarrow^* P(\alpha2, \beta1, \gamma2, \beta2, n, 1) \rightarrow P(\alpha2, \beta2, \gamma2, \beta1, 1, 1) \rightarrow^* P(\alpha2, \beta2, \gamma2, x2, y2, z2)$.

c) A detailed description of the path used is as follows:

$P(\alpha1, \beta1, \gamma1, x1, y1, z1) \rightarrow P(\alpha1, \beta1, \gamma1, x1 + 1, y1, z1) \rightarrow \ldots \rightarrow P(\alpha1, \beta1, \gamma1, n, y1, z1) \rightarrow P(\alpha1, \beta1, \gamma1, n, y1, z1 + 1) \rightarrow \ldots \rightarrow P(\alpha1, \beta1, \gamma1, n, y1, \alpha2) \rightarrow P(\alpha2, \beta1, \gamma1, 1, y1, \alpha1) \rightarrow P(\alpha2, \beta1, \gamma1, 1, y1, \alpha1+1) \rightarrow \ldots \rightarrow P(\alpha2, \beta1, \gamma1, 1, y1, n)$

$\rightarrow P(\alpha 2, \beta 1, \gamma 1, 1, y1 + 1, n) \rightarrow ... \rightarrow P(\alpha 2, \beta 1, \gamma 1, 1, \gamma 2, n) \rightarrow P(\alpha 2, \beta 1, \gamma 2, 1, \gamma 1, 1) \rightarrow P(\alpha 2, \beta 1, \gamma 2, 1, \gamma 1 + 1, 1) \rightarrow ... \rightarrow P(\alpha 2, \beta 1, \gamma 2, 1, n, 1) \rightarrow P(\alpha 2, \beta 1, \gamma 2, 1 + 1, n, 1) \rightarrow ... \rightarrow P(\alpha 2, \beta 1, \gamma 2, \beta 2, n, 1) \rightarrow P(\alpha 2, \beta 2, \gamma 2, \beta 1, 1, 1) \rightarrow P(\alpha 2, \beta 2, \gamma 2, \beta 1 - 1, 1, 1) \rightarrow ... \rightarrow P(\alpha 2, \beta 2, \gamma 2, x2, 1, 1) \rightarrow P(\alpha 2, \beta 2, \gamma 2, x2, 1, 1 + 1) \rightarrow ... \rightarrow P(\alpha 2, \beta 2, \gamma 2, x2, 1, z2) \rightarrow P(\alpha 2, \beta 2, \gamma 2, x2, 1+1, z2) \rightarrow ... \rightarrow P(\alpha 2, \beta 2, \gamma 2, x2, y2, z2).$

Path EDPT3:

a) At the block level the path is as follows:

$B(\alpha 1, \beta 1, \gamma 1) \rightarrow B(\alpha 1, \beta 2, \gamma 1) \rightarrow B(\alpha 2, \beta 2, \gamma 1) \rightarrow B(\alpha 2, \beta 2, \gamma 2)$

b) A short description of the path is as follows:

$P(\alpha 1, \beta 1, \gamma 1, x1, y1, z1) \rightarrow P(\alpha 1, \beta 1, \gamma 1, \beta 2, 1, z1) \rightarrow P(\alpha 1, \beta 2, \gamma 1, \beta 1, n, z1) \rightarrow P(\alpha 1, \beta 2, \gamma 1, n, n, \alpha 2) \rightarrow P(\alpha 2, \beta 2, \gamma 1, 1, n, \alpha 1) \rightarrow P(\alpha 2, \beta 2, \gamma 1, 1, \gamma 2, n) \rightarrow P(\alpha 2, \beta 2, \gamma 2, 1, \gamma 1, 1) \rightarrow P(\alpha 2, \beta 2, \gamma 2, x2, y2, z2).$

c) A detailed description of the path used is as follows

$P(\alpha 1, \beta 1, \gamma 1, x1, y1, z1) \rightarrow P(\alpha 1, \beta 1, \gamma 1, x1, y1 - 1, z1) \rightarrow ... \rightarrow P(\alpha 1, \beta 1, \gamma 1, x1, 1, z1) \rightarrow P(\alpha 1, \beta 1, \gamma 1, x1 + 1, 1, z1) \rightarrow ... \rightarrow P(\alpha 1, \beta 1, \gamma 1, \beta 2, 1, z1) \rightarrow P(\alpha 1, \beta 2, \gamma 1, \beta 1, n, z1) \rightarrow P(\alpha 1, \beta 2, \gamma 1, \beta 1 + 1, n, z1) \rightarrow ... \rightarrow P(\alpha 1, \beta 2, \gamma 1, n, n, z1) \rightarrow P(\alpha 1, \beta 2, \gamma 1, n, n, z1 + 1) \rightarrow ... \rightarrow P(\alpha 1, \beta 2, \gamma 1, n, n, \alpha 2) \rightarrow P(\alpha 2, \beta 2, \gamma 1, 1, n, \alpha 1) \rightarrow P(\alpha 2, \beta 2, \gamma 1, 1, n - 1, \alpha 1) \rightarrow ... \rightarrow P(\alpha 2, \beta 2, \gamma 1, 1, \gamma 2, \alpha 1) \rightarrow P(\alpha 2, \beta 2, \gamma 1, 1, \gamma 2, \alpha 1 + 1) \rightarrow ... \rightarrow P(\alpha 2, \beta 2, \gamma 1, 1, \gamma 2, n) \rightarrow P(\alpha 2, \beta 2, \gamma 2, 1, \gamma 1, 1) \rightarrow P(\alpha 2, \beta 2, \gamma 2, 1, \gamma 1 - 1, 1) \rightarrow ... \rightarrow P(\alpha 2, \beta 2, \gamma 2, 1, y2, 1) \rightarrow P(\alpha 2, \beta 2, \gamma 2, 1, y2, 1 + 1) \rightarrow ... \rightarrow P(\alpha 2, \beta 2, \gamma 2, 1, y2, z2) \rightarrow P(\alpha 2, \beta 2, \gamma 2, 1+1, y2, z2) \rightarrow ... \rightarrow P(\alpha 2, \beta 2, \gamma 2, x2, y2, z2).$

Path EDPT4:

a) At the block level the path is as follows:

B (α1, β1, γ1) → B (α1, β2, γ1) → B (α1, β2, γ2) → B (α2, β2, γ2)

b) A short description of the path is as follows:

P(α1, β1, γ1, x1, y1, z1) → P(α1, β1, γ1, β2, *n*, z1) → P(α1, β2, γ1, β1, 1,z1) → P(α1, β2, γ1,β1 , γ2, 1) → P(α1, β2, γ2, β1, γ1, *n*) → P(α1, β2, γ2, *n*, 1, α2) → P(α2, β2, γ2, 1 ,1, α1) → P(α2, β2, γ2 , x2, y2, z2).

c) A detailed description of the path used is as follows

P (α1, β1, γ1, x1, y1, z1) → P (α1, β1, γ1, x1, y1 + 1, z1) → ... → P(α1, β1, γ1, x1, *n*, z1) → P(α1, β1, γ1, x1 + 1, *n*, z1) → ... → P(α1, β1, γ1, β2, *n*, z1) → P(α1, β2, γ1, β1, 1, z1) → P(α1, β2, γ1, β1, 1 + 1, z1) → ... → P(α1, β2, γ1, β1 , γ2, z1) → P(α1, β2, γ1, β1, γ2, z1 -1) → ... → P(α1, β2, γ1, β1, γ2, 1) → P(α1, β2, γ2, β1, γ1, *n*) → P(α1, β2, γ2, β1 - 1, γ1, *n*) → ... → P(α1, β2, γ2, 1, γ1, *n*) → P(α1, β2, γ2, 1, γ1 – 1, *n*) → ... → P(α1, β2, γ2, 1, 1, *n*) → P(α1, β2, γ2, 1, 1, *n* - 1) → ... → P(α1, β2, γ2, 1, 1, α2) → P(α2, β2, γ2, *n* ,1, α1) → P(α2, β2, γ2, *n*,1, α1 - 1) → ... → P(α2, β2, γ2, *n* ,1, z2) → P(α2, β2, γ2, *n* ,1 +1, z2) → ... → P(α2, β2, γ2, *n* , y2, z2) → P(α2, β2, γ2, *n-1*, y2, z2) → ... → P(α2, β2, γ2 , x2, y2, z2).

Path EDPT5:

a) At the block level the path is as follows:

B (α1, β1, γ1) → B (α1, β1, γ2) → B (α1, β2, γ2) → B (α2, β2, γ2)

b) A short description of the path is as follows:

P(α1, β1, γ1, x1, y1, z1) → P(α1, β1, γ1, x1, γ2, 1) → P(α1, β1, γ2, x1, γ1, *n*) → P(α1, β1, γ2, β2, *n*, *n*) → P(α1, β2, γ2, β1, 1, *n*) → P(α1, β2, γ2, *n*, 1, α2) → P(α2, β2, γ2, 1 ,1, α1) → P(α2, β2, γ2 , x2, y2, z2).

c) A detailed description of the path used is as follows

$P(\alpha1, \beta1, \gamma1, x1, y1, z1) \rightarrow P(\alpha1, \beta1, \gamma1, x1, y1, z1 - 1) \rightarrow ... \rightarrow P(\alpha1, \beta1, \gamma1, x1, y1, 1) \rightarrow P(\alpha1, \beta1, \gamma1, x1, y1 + 1, 1) \rightarrow ... \rightarrow P(\alpha1, \beta1, \gamma1, x1, \gamma2, 1) \rightarrow P(\alpha1, \beta1, \gamma2, x1, \gamma1, n) \rightarrow P(\alpha1, \beta1, \gamma2, x1, \gamma1 - 1, n) \rightarrow ... \rightarrow P(\alpha1, \beta1, \gamma2, x1,1, n) \rightarrow P(\alpha1, \beta1, \gamma2, x1 + 1, 1, n) \rightarrow ... \rightarrow P(\alpha1, \beta1, \gamma2, \beta2, 1, n) \rightarrow P(\alpha1, \beta2, \gamma2, \beta1, n, n) \rightarrow P(\alpha1, \beta2, \gamma2, \beta1 + 1, n, n) \rightarrow ... \rightarrow P(\alpha1, \beta2, \gamma2, n, n, n) \rightarrow P(\alpha1, \beta2, \gamma2, n, n, n - 1) \rightarrow ... \rightarrow P(\alpha1, \beta2, \gamma2, n, n, \alpha2) \rightarrow P(\alpha2, \beta2, \gamma2, 1, n, \alpha1) \rightarrow P(\alpha2, \beta2, \gamma2, 1, n, \alpha1 - 1) \rightarrow ... \rightarrow P(\alpha2, \beta2, \gamma2, 1, n, z2) \rightarrow P(\alpha2, \beta2, \gamma2, 1 + 1, n, z2) \rightarrow ... \rightarrow P(\alpha2, \beta2, \gamma2, x2, n, z2) \rightarrow P(\alpha2, \beta2, \gamma2, x2, n- 1, z2) \rightarrow ... \rightarrow P(\alpha2, \beta2, \gamma2, x2, y2, z2)$.

Path EDPT6:

a) At the block level the path is as follows:

$B(\alpha1, \beta1, \gamma1) \rightarrow B(\alpha1, \beta1, \gamma2) \rightarrow B(\alpha2, \beta1, \gamma2) \rightarrow B(\alpha2, \beta2, \gamma2)$

b) A short description of the path is as follows:

$P(\alpha1, \beta1, \gamma1, x1, y1, z1) \rightarrow P(\alpha1, \beta1, \gamma1, x1, \gamma2, n) \rightarrow P(\alpha1, \beta1, \gamma2, x1, \gamma1, 1) \rightarrow P(\alpha1, \beta1, \gamma2, 1, n, \alpha2) \rightarrow P(\alpha2, \beta1, \gamma2, n, n, \alpha1) \rightarrow P(\alpha2, \beta1, \gamma2, \beta2, n, n) \rightarrow P(\alpha2, \beta2, \gamma2, \beta1, 1, n) \rightarrow P(\alpha2, \beta2, \gamma2, x2, y2, z2)$.

c) A detailed description of the path used is as follows

$P(\alpha1, \beta1, \gamma1, x1, y1, z1) \rightarrow P(\alpha1, \beta1, \gamma1, x1, y1, z1 + 1) \rightarrow ... \rightarrow P(\alpha1, \beta1, \gamma1, x1, y1, n) \rightarrow P(\alpha1, \beta1, \gamma1, x1, y1 + 1, n) \rightarrow ... \rightarrow P(\alpha1, \beta1, \gamma1, x1, \gamma2, n) \rightarrow P(\alpha1, \beta1, \gamma2, x1, \gamma1, 1) \rightarrow P(\alpha1, \beta1, \gamma2, x1 - 1, \gamma1, 1) \rightarrow ... \rightarrow P(\alpha1, \beta1, \gamma2, 1, \gamma1, 1) \rightarrow P(\alpha1, \beta1, \gamma2, 1, \gamma1 + 1, 1) \rightarrow ... \rightarrow P(\alpha1, \beta1, \gamma2, 1, n, 1) \rightarrow P(\alpha1, \beta1, \gamma2, 1, n, 1 + 1) \rightarrow ... \rightarrow P(\alpha1, \beta1, \gamma2, 1, n, \alpha2) \rightarrow P(\alpha2, \beta1, \gamma2, n, n, \alpha1) \rightarrow P(\alpha2, \beta1, \gamma2, n, n, \alpha1 + 1) \rightarrow ... \rightarrow P(\alpha2, \beta1, \gamma2, n, n, n) \rightarrow P(\alpha2, \beta1, \gamma2, n - 1, n, n) \rightarrow ... \rightarrow P(\alpha2, \beta1, \gamma2, \beta2, n, n) \rightarrow P(\alpha2, \beta2, \gamma2, \beta1, 1, n) \rightarrow P(\alpha2, \beta2, \gamma2, \beta1 - 1, 1, n) \rightarrow ... \rightarrow P(\alpha2, \beta2, \gamma2, x2, 1, n) \rightarrow P(\alpha2, \beta2, \gamma2, x2, 1 + 1, n) \rightarrow ... \rightarrow P(\alpha2, \beta2, \gamma2, x2, y2, n) \rightarrow P(\alpha2, \beta2, \gamma2, x2, y2, n - 1) \rightarrow ... \rightarrow P(\alpha2, \beta2, \gamma2, x2, y2, z2)$.

It may be readily verified that these six paths are edge-disjoint. The other cases are similar.

3.3 Message Routing in the 3D Multi-Mesh

Routing problem can be defined as the process of sending messages from source processor to destination processor. The routing algorithm implemented on the router is responsible for determining the path from source to destination. The length of the path in the worst possible situation determines the performance of a routing algorithm.

In this section we present routing messages from any source processor to any destination processor for point to point communication. Let the source processor be **S** = P(α1, β1, γ1, x1, y1, z1) and the destination processor be **D** = P(α2, β2, γ2, x2, y2, z2), $1 \leq \alpha 1, \beta 1, \gamma 1, x1, y1, z1, \alpha 2, \beta 2, \gamma 2, x2, y2, z2 \leq n$, so that the 3D block corresponding to **S** is B(α1, β1, γ1) and that corresponding to **D** is B(α2, β2, γ2). We will describe the routing along the restricted path such as the Theorem 3.1 (Diameter). There are three situation to consider that we describe in Theorem 3.1, among them we'll only consider the situation where none of the coordinates of the source block B (α1, β1, γ1) have the same value as that of the corresponding coordinate in the destination block B (α2, β2, γ2). The other cases are similar.

In this case, none of the coordinates of the source block B (α1, β1, γ1) have the same value as that of the corresponding coordinate in the destination block B (α2, β2, γ2). In

other words, $\alpha 1 \neq \alpha 2$, $\beta 1 \neq \beta 2$ and $\gamma 1 \neq \gamma 2$. In this case, there exist two intermediate blocks B ($\alpha 3$, $\beta 3$, $\gamma 3$) and B ($\alpha 4$, $\beta 4$, $\gamma 4$) such that there is a direct link between

- the source block B($\alpha 1$, $\beta 1$, $\gamma 1$) and the intermediate block B($\alpha 3$, $\beta 3$, $\gamma 3$),
- the block B($\alpha 3$, $\beta 3$, $\gamma 3$) and the block B($\alpha 4$, $\beta 4$, $\gamma 4$) and
- the block B($\alpha 4$, $\beta 4$, $\gamma 4$) and the destination block B($\alpha 2$, $\beta 2$, $\gamma 2$).

There are a number of ways in which we may choose the intermediate blocks B ($\alpha 3$, $\beta 3$, $\gamma 3$) and B ($\alpha 4$, $\beta 4$, $\gamma 4$). For example, we could select B ($\alpha 2$, $\beta 1$, $\gamma 1$) and B ($\alpha 2$, $\beta 1$, $\gamma 2$) as intermediate nodes. In order to route a message from a source processor **S** = ($\alpha 1$, $\beta 1$, $\gamma 1$, x1, y1, z1) to any destination processor **D** = P($\alpha 2$, $\beta 2$, $\gamma 2$, x2, y2, z2), we first divide the source block by three imaginary planes - ($\alpha 1$, $\beta 1$, $\gamma 1$, $\beta 2$, *, *), ($\alpha 1$, $\beta 1$, $\gamma 1$, *, $\gamma 2$, *) and ($\alpha 1$, $\beta 1$, $\gamma 1$, *, *, $\alpha 2$) that we showed in figure 3.5. This gives us 8 octants in the source block, which we will denote by SO1, SO2, SO3, SO4, SO5, SO6, SO7 and SO8. Similarly we divide the destination block by three other imaginary planes ($\alpha 2$, $\beta 2$, $\gamma 2$, $\beta 1$, *, *), ($\alpha 2$, $\beta 2$, $\gamma 2$, *, $\gamma 1$, *) and ($\alpha 2$, $\beta 2$, $\gamma 2$, *, *, $\alpha 1$) giving us 8 octants- DO1, DO2, DO3, DO4, DO5, DO6, DO7 and DO8 in the destination block.

We will use the boundary processors for the three planes described as exit/entry points to communicate to processors in other blocks. In the proof for theorem 3.1, we showed that, for a suitable choice of the exit point from the source block, we could choose a corresponding entry point for the destination block to define a path PT1. Keeping in mind the choices for the entry/exit points for PT1, we chose another set of entry/exit points to define a path PT2. We showed that one of these paths must be of length less than or equal

to $3n$. The algorithm for message routing uses this idea to select the optimum path for routing.

The idea used in this algorithm is similar to that used in [DDS99].

Algorithm:

Step 1:

i) Determine the octets of the source and destination blocks.

ii) Calculate the two possible paths PT1 and PT2 from source processor to destination processor as defined in section 3 and choose the path with the shortest length. Let the chosen path from $B(\alpha 1, \beta 1, \gamma 1)$ to $B(\alpha 2, \beta 2, \gamma 2)$ be through blocks $B(\alpha_{i1}, \beta_{j1}, \gamma_{k1})$ and $B(\alpha i_2, \beta_{j2}, \gamma_{k2})$.

iii) Attach, to the data packet, a list consisting of the addresses of the exit/entry processors of these blocks. This list consists of the following pieces of information:

Field1: Source block exit processor

Field2: First intermediate block exit processor

Field3: Second intermediate block exit processor and

Field4: Destination processor entry processor.

Step 2:

If the value stored in Field1 is the address of the current processor, go to step 3. Otherwise send the packet towards the processor specified in Field1 using the appropriate intra-block link from the current processor and go back to step 2.

Step 3:

a) Send the packet to the appropriate processor by using appropriate inter-block link from the current processor.

b) Update the list of four address field information as follows-

Field1 ← **Field2**

Field2 ← **Field3**

Field3 ← **Field4**

Field4 ← NULL

c) If Field1 is NULL, stop. Otherwise, and go to step 2.

Example:

A possible path PT1 from the source processor (which is in the block ($\alpha1$, $\beta1$, $\gamma1$)) to the destination processor (in the block ($\alpha2$, $\beta2$, $\gamma2$)) using the intermediate blocks ($\alpha2$, $\beta1$, $\gamma1$) and ($\alpha2$, $\beta2$, $\gamma1$) may be formulated as follows-

PT1 = P($\alpha1$, $\beta1$, $\gamma1$, x1, y1, z1) → ... → P($\alpha1$, $\beta1$, $\gamma1$, 1, y1, $\alpha2$) → P($\alpha2$, $\beta1$, $\gamma1$, *n*, y1, $\alpha1$) →... → P($\alpha2$, $\beta1$, $\gamma1$, $\beta2$, 1, $\alpha1$) → P($\alpha2$, $\beta2$, $\gamma1$, $\beta1$, *n* ,$\alpha1$) → ... → P($\alpha2$, $\beta2$, $\gamma1$, $\beta1$, $\gamma2$, *n*) → P($\alpha2$, $\beta2$, $\gamma2$, $\beta1$, $\gamma1$, 1) → ... → P($\alpha2$, $\beta2$, $\gamma2$, x2, y2, z2).

Initially data packet appends the following four fields-

1. ($\alpha1$, $\beta1$, $\gamma1$, 1, y1, $\alpha2$) as **Field1**
2. ($\alpha2$, $\beta1$, $\gamma1$, $\beta2$, 1, $\alpha1$) as **Field2**
3. ($\alpha2$, $\beta2$, $\gamma1$, $\beta1$, $\gamma2$, *n*) as **Field3**
4. ($\alpha2$, $\beta2$, $\gamma2$, x2, y2, z2) as **Field4**

Source processor first checks Field1, if the address of Field1 is not the current processor then routes the messages to the processor P ($\alpha1$, $\beta1$, $\gamma1$, 1, y1, $\alpha2$) via an intra-block link.

If the address of Field1 is the current processor then route the message to the processor P($\alpha2$, $\beta1$, $\gamma1$, *n*, y1, $\alpha1$) by using the inter-block link and updates the list of four fields as follows-

Field1 ← ($\alpha2$, $\beta1$, $\gamma1$, $\beta2$, 1, $\alpha1$)

Field2 ← ($\alpha2$, $\beta2$, $\gamma1$, $\beta1$, $\gamma2$, *n*)

Field3 ← (α2, β2, γ2 , x2, y2, z2)

Field4 ← NULL

Then the processor P(α2, β1, γ1, *n*, y1, α1) checks the Field1 and route the messages to the processor (Field1) P (α2, β1, γ1, β2, 1, α1) by using intra-block link. Processor P (α2, β1, γ1, β2, 1, α1) has the same address as Field1 so it routes message to processor P (α2, β2, γ1, β1, *n*, α1) via the inter-block link and updates the four fields as follows-

Field1 ← (α2, β2, γ1, β1, γ2, *n*)

Field2 ← (α2, β2, γ2, x2, y2, z2)

Field3 ← NULL

Field4 ← NULL

Processor P (α2, β2, γ1, β1, *n*, α1) checks the Field1 and route the messages to the processor (Field1) that is P (α2, β2, γ1, β1, γ2, *n*) by using intra-block link. Processor P(α2, β2, γ1, β1, γ2, *n*) sends the message to P(α2, β2, γ2, β1, γ1, 1) by using inter-block link and updates the four fields as follows-

Field1 ← (α2, β2, γ2, x2, y2, z2)

Field2 ← NULL

Field3 ← NULL

Field4 ← NULL

This processor P (α2, β2, γ2, β1, γ1, 1) checks the Field1 and route the messages to the address of Field1 that is (α2, β2, γ2, x2, y2, z2) by using intra-block link and updates the four fields as follows-

Field1 ← NULL

Field2 ← NULL

Field3 ← NULL

Field4 ← NULL

As the value of the field1 is sets to NULL the routing process is terminated.

3. 4 Summation/Average/Minimum/Maximum in the 3D Multi-Mesh

We may use the 3D MM to compute the sum of up to n^6 data values stored in the n^6 processors of a 3D MM of order n. The same idea may be used to compute the average, maximum or minimum of up to n^6 data values. The scheme we use is similar to that used in [DDS99] for the Multi-Mesh. We assume that each processor has three registers X, Y and Z for data communication in the three axes and will use X (α, β, γ, x, y, z) (Y (α, β, γ, x, y, z) and Z (α, β, γ, x, y, z)) to denote the X (respectively Y and Z) register in processor P (α, β, γ, x, y, z). The data is initially in register Z of all n^6 processors in the 3D MM. The main idea of the algorithm, is to

i) compute, in parallel, the sum of all numbers in each 3D block,
ii) communicate the partial sums to blocks B(1, β, γ), $1 \leq \beta, \gamma \leq n$,
iii) compute the sum of the partial sum of all numbers in B(1, β, γ), $1 \leq \beta, \gamma \leq n$ and communicate the partial sums to blocks B(1, 1, γ), $1 \leq \gamma \leq n$,
iv) compute the sum of the partial sum of all numbers in B(1, 1, γ), $1 \leq \gamma \leq n$ and communicate the result to block B(1, 1, 1).

The algorithm is as follows:

Algorithm Sum

Step 1

$\forall \alpha, \beta, \gamma, x, y, 1 \leq \alpha, \beta, \gamma, x, y \leq n$ do in parallel

for k = n - 1 downto 1 do

$Z(\alpha, \beta, \gamma, x, y, k) \leftarrow Z(\alpha, \beta, \gamma, x, y, k + 1) + Z(\alpha, \beta, \gamma, x, y, k)$;

/* Z(α, β, γ, x, y, 1) now contains the partial sum of n values */

Y(α, β, γ, x, y, 1) ← Z(α, β, γ, x, y, 1);

for j = n - 1 downto 1 do

Y(α, β, γ, x, j, 1) ← Y(α, β, γ, x, j+ 1, 1) + Y(α, β, γ, x, j, 1);

/* Y(α, β, γ, x, 1, 1) now contains the partial sum of n^2 values */

X(α, β, γ, x, 1, 1) ← Y(α, β, γ, x, 1, 1);

for i = n - 1 downto 1 do

X(α, β, γ, i, 1, 1) ← X(α, β, γ, i+1, 1, 1) + X(α, β, γ, i, 1, 1);

/* X(α, β, γ, 1, 1, 1) now contains the partial sum of n^3 values */

Y(α, β, γ, 1, 1, 1) ← X(α, β, γ, 1, 1, 1);

Y(α, β, 1, 1, γ, n) ← Y(α, β, γ, 1, 1, 1);

/*Using the link **(γ ↔ y)** the partial sums in blocks B(α, β, *) are transferred to blocks B(α, β, 1) */

Step 2

∀α, β, 1 ≤ α, β, ≤ n do in parallel

for j = n - 1 downto 1 do

Y(α, β, 1, 1, j, n) ← Y(α, β, 1, 1, j+ 1, n) + Y(α, β, 1, 1, j, n);

/* Y(α, β, 1, 1, 1, n) now contains the partial sum of n^4 values */

Y(α, β, 1, 1, 1, 1) ← Y(α, β, 1, 1, 1, n); /*Using the link **(γ ↔ y)** */

X(α, β, 1, 1, 1, 1) ← Y(α, β, 1, 1, 1, 1);

X(α, 1, 1, β, n, 1) ← X(α, β, 1, 1, 1, 1); /* Using the link **(β ↔ x)**/

Step 3

∀α, 1 ≤ α, ≤ n do in parallel

for i = n - 1 downto 1 do

$X(\alpha, 1, 1, i, n, 1) \leftarrow X(\alpha, 1, 1, i+1, n, 1) + X(\alpha, 1, 1, i, n, 1)$;

/* $X(\alpha, 1, 1, 1, n, 1)$ now contains the partial sum of n^5 values */

$X(\alpha, 1, 1, 1, 1, 1) \leftarrow X(\alpha, 1, 1, 1, n, 1)$; /*Using the link $(\beta \leftrightarrow x)$*/

$Z(\alpha, 1, 1, 1, 1, 1) \leftarrow X(\alpha, 1, 1, 1, 1, 1)$;

$Z(1, 1, 1, n, 1, \alpha) \leftarrow Z(\alpha, 1, 1, 1, 1, 1)$; /* Using the link $(\alpha \leftrightarrow z)$*/

Step 4

for k = n - 1 downto 1 do

$Z(1, 1, 1, n, 1, k) \leftarrow Z(1, 1, 1, n, 1, k+1) + Z(1, 1, 1, n, 1, k)$;

/* $Z(1, 1, 1, n, 1, 1)$ now contains the partial sum of n^6 values */

$Z(1, 1, 1, 1, 1, 1) \leftarrow Z(1, 1, 1, n, 1, 1)$; /*Using the link $(\alpha \leftrightarrow z)$*/

We will now analyze the time needed for this algorithm. Let t_c denote the time for one communication, assuming that inter-block and intra-block communication take the same time and t_a denote the time for one addition. In the steps where we have use an addition(+) operation, the operation is actually one data communication and one addition so that the total time needed for the operation is $t_c + t_a$. Step 1 takes $n(t_c + t_a) + t_c + n(t_c + t_a) + t_c + n(t_c + t_a) + t_c = (3n + 3)\, t_c + 3n\, t_a$ time units. Step 2 takes $n\,(t_c + t_a) + 2t_c = (n + 2)\, t_c + n\, t_a$ time units. Step 3 takes $n\,(t_c + t_a) + 2t_c = (n + 2)\, t_c + n\, t_a$ time units. Step 4 takes $n\,(t_c + t_a) + t_c = (n + 1)\, t_c + n\, t_a$ time units. The total time required is $(6n + 8)\, t_c + 6n\, t_a$ time. Thus the algorithm to compute the sum of n^6 numbers on the 3D MM is O (n). This may be compared to the time O (n) to compute the sum of n^4 numbers on the Multi-Mesh.

In this chapter we have defined the 3D Multi-Mesh architecture, studied the diameter and connectivity of this network, and have developed two important algorithms for the 3D MM.

Chapter 4

Optical Implementation of Multi-Mesh Links

In order to improve the performance of multiprocessor systems, the use of very high-speed communication technology is crucial. As mentioned in chapter 2, to avoid the limitations of electronic technology that uses copper, optical technology is promising for inter-processor communication. We have reviewed a number of optics-based interconnection schemes as well as hybrid (electronic and optical) schemes in chapter 2. To our knowledge there is no research on implementing the Multi-Mesh using optical technology. In this chapter we will discuss how the Multi-Mesh architecture may be implemented using optical technology and we have described a number of possible approaches for designing optics-based interconnections for the Multi-Mesh. Our results may be extended to define 3D Multi-Mesh using optical technology.

We have already mentioned that the Multi-Mesh (MM) network discussed in Chapter 2 has attractive topological attributes. In a MM network of order n, there are n^2 blocks (where a block is a mesh of processors) arranged in the form of an $n \times n$ matrix. In chapter 2, Figure 2.11 shows a MM of order 3. In a MM network, the processors within a block are connected by intra-block links to other processors in the same block. Some processors of different blocks are connected by inter-block links. We now show how we can implement the inter-block connections of the MM network by using the wavelength routed WDM technology we also reviewed in chapter 2. To present our design, it is

convenient for us to separate the intra-block connections from the inter-block connections. We show a MM network of order 4 in Figure 4.1 where each square represents a two-dimensional mesh and inter-block connections are omitted. We show a single 2-dimensional block of order 4 in Figure 4.2 where each circle represents a processor and each edge represents an intra-block connection.

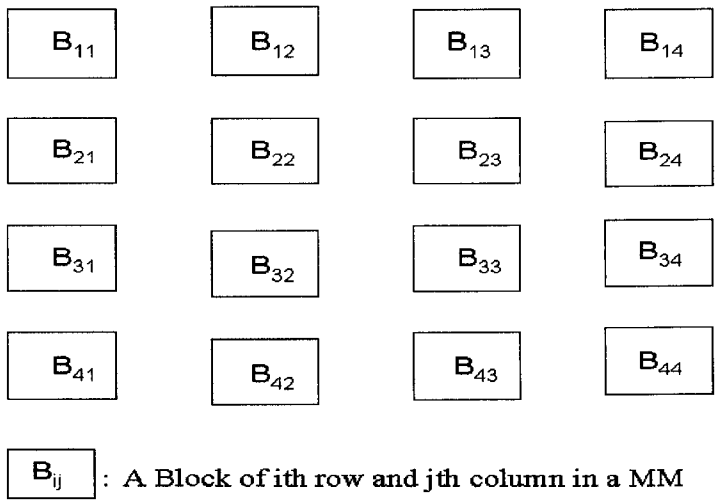

Figure 4.1: 4 X 4 Blocks of a Multi-Mesh network of order 4

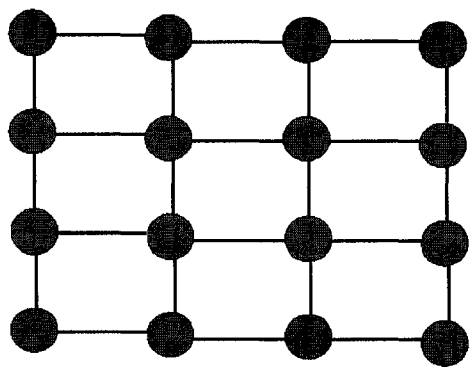

Figure 4.2: A block of a Multi-Mesh network of order 4

We note that intra-block connections are short as compared to the length of inter-block connections and have a constant length. It is convenient to implement such connections using VLSI technology or by using free space optical communication. In this thesis we will only look at a hybrid approach where we use metal lines when fabricating the array of processors using VLSI technology. The alternative of using free space optical communication as proposed in [LoSu94b] is quite straightforward.

The more challenging task is to realize the inter-block connections since the length of such a connection changes and becomes very long for large networks.

There are two novel and interesting features of our implementation of inter-block connections:

- The first attractive feature is that we have used wavelength routed WDM networks rather than WDM networks based on passive star couplers [Mu97]. It is well known that the power requirements for passive star couplers make them unsuitable for large networks [Mu97]. In other words, using our approach, we can easily define larger networks with a relatively lower power budget.

- The second interesting feature is that we have incorporated fault tolerance using protection scheme [Ge98], [RaMu99b], [SRM02]. The idea is that each pair of processors that are connected by an optical link will have 2 edge-disjoint optical paths - the primary path and the back-up path. If there is a failure in the primary path, simply the router settings have to be changed so that the back up path can

be used. This means that, in the case of a single failure in the optical part of the network, the overall routing scheme does not have to be changed and the network diameter is not affected.

To our knowledge no other interconnection network has used these two ideas.

To realize the inter-block connections, our tasks are to

- define a physical topology consisting of fibers, routers and end-nodes (the sources or destinations of data). In the case of a Multi-Mesh, the end-nodes are the boundary processors of each block in the Multi-Mesh.

- define a logical topology on the physical topology such that for every undirected inter-block link between x and y in a Multi-Mesh there is a logical edge $x \rightarrow y$ and a logical edge $y \rightarrow x$ in the logical topology. For economic reasons, we wish to use as few wavelengths as possible.

In a wavelength-routed network given a physical topology, in order to define a logical topology, we have to

- determine which processors need to be connected by a lightpath,
- determine a viable route and a wavelength for each lightpath (RWA problem) [Mu97], [StBa99] .

Since we are implementing a known pattern of connections (as defined by the inter-block connection rules of the Multi-Mesh), the lightpaths we need are already defined. In the following sections, we first discuss possible physical topologies for this problem and present ways to handle the RWA problem to realize the desired connections for the fault-free and faulty situations that we have considered.

4.1 Physical Topology for Optical Communication in a Multi-Mesh

In our scheme we propose to use n^2 routers - one for each of the n^2 blocks. Figure 4.3 shows part of our physical topology where a square represents a block (which, as explained earlier, is a mesh of processors) and an oval represents an optical router. All the routers are arranged in the form of a two-dimensional grid. To simplify the diagram we have not shown the connections from the boundary processors to the routers. As shown in Figure 4.3, the connection between the routers is the architecture of a torus. For clarity, we have shown the wrap-around links only for the first and the last rows and columns. Each row and column has similar connections.

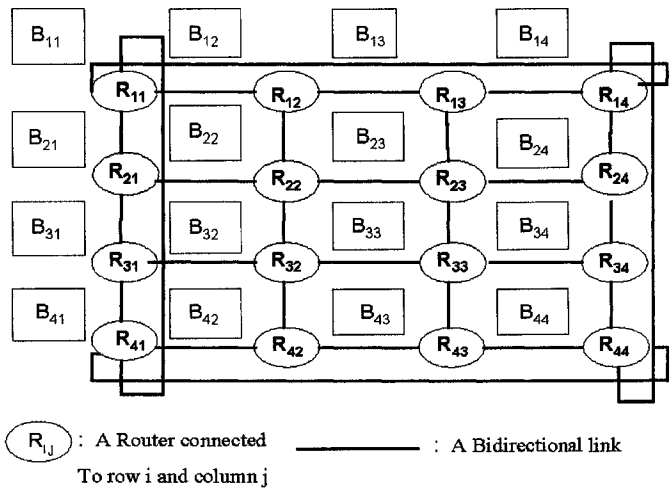

Figure 4.3: Connections between Routers in a Multi-Mesh network of order 4

At this point, in Figure 4.3, we have used undirected links. Later on, we will implement such links using either unidirectional links or bi-directional links. If there is a unidirectional link x $\rightarrow$ y, it means there is a fiber allowing communication from node x to node y. It is not necessarily true that there will be a fiber allowing communication from node y to node x. In the case of bi-directional link x $\leftrightarrow$ y, there will always be two fibers - one allowing communication from x to y and one for communication from y to x.

Now we will discuss how we propose to connect the boundary processors of a block to a router. We will discuss in detail the physical topology corresponding to the connections from the boundary processors on the top and the bottom edge of block B_{ij}. The physical topology corresponding to the connections from the boundary processors on the right and the left edge of block B_{ij} are similar.

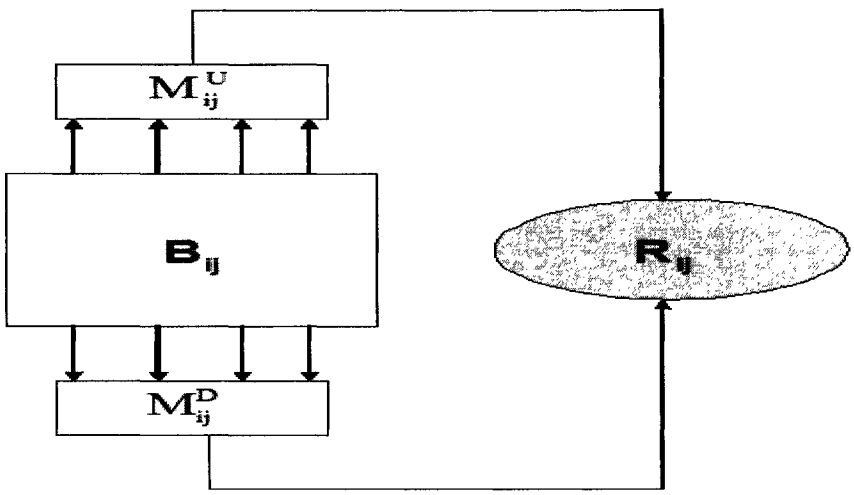

Figure 4.4: Outputs of multiplexers are connected to the inputs of router

Router R_{ij} will be connected to the corresponding block B_{ij} carrying incoming and outgoing optical signals as follows:

1) the router R_{ij} will be connected to block B_{ij} with one fiber carrying signals from processors P(i, j, 1, k) of block B_{ij} for communication to processor P(k, j, n, i) of block B_{kj}, for all k, $1 \leq k \leq n$, $k \neq j$. This may be easily achieved by using a multiplexer M_{ij}^{U}, shown in Figure 4.4 with inputs from processors P(i, j, 1, k), for all k, $1 \leq k \leq n$. The fiber carrying the output of multiplexer M_{ij}^{U} is connected as an input to router R_{ij} as shown in Figure 4.4. We will later use this fiber to define logical edges corresponding to the inter-block connections from the first row of the block B_{ij} to the n^{th} row of the other blocks in the same column.

2) the router R_{ij} will be connected to block B_{ij} with one fiber carrying signals from processors P(i, j, n, k) of block B_{ij} to processor P(k, j, 1, i) of block B_{kj}, for all k, $1 \leq k \leq n$, $k \neq j$. This may be easily achieved by using a multiplexer M_{ij}^{D} shown in Figure 4.4 with inputs from processors P (i, j, n, k), for all k, $1 \leq k \leq n$. The fiber carrying the output of multiplexer M_{ij}^{D} is connected an input to router R_{ij} as shown in Figure 4.4. We will later use this fiber to define logical edges corresponding to the inter-block connections from the n^{th} row of the block B_{ij} to the first row of the other blocks in the same column.

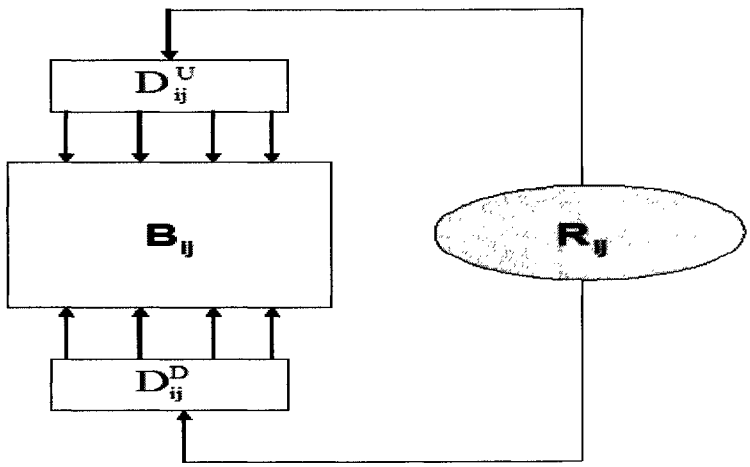

Figure 4.5: Inputs of the demultiplexers are connected to the output of router

3) the router R_{ij} will be connected to block B_{ij} with one fiber carrying signals from processors P(k, j, *n*, i) of block B_{kj} to processor P(i, j, 1, k) of block B_{ij}, for all k, $1 \leq k \leq n$, $k \neq j$. This may be easily achieved by using a de-multiplexer D_{ij}^{U}, shown in Figure 4.5 with inputs from processors P(k, j, n, i) for all k, $1 \leq k \leq n$. The fiber carrying the input to de-multiplexer D_{ij}^{U} is an output from the router R_{ij} as shown in Figure 4.5. We will later use this fiber to define the logical edges corresponding to the inter-block connections from the n^{th} row of the blocks B_{kj} to the first row of the block B_{ij} in the same column.

4) the router R_{ij} will be connected to block B_{ij} with one fiber carrying signals from processors P(k, j, 1, i) of block B_{kj} to processor P(i, j, *n*, k) of block B_{ij}, for all k, $1 \leq k \leq n$, $k \neq j$. This may be easily achieved by using a de-multiplexer D_{ij}^{D}, shown in Figure 4.5 with inputs from processors P(k, j, *n*, i) for all k, $1 \leq k \leq n$. The fiber carrying the input to the de-multiplexer D_{ij}^{D} is an output to router R_{ij} as

shown in Figure 4.5. We will later use this fiber to define the logical edges corresponding to the inter-block connections from the first row of other blocks B_{kj} to the n^{th} row of the block B_{ij} in the same column.

The Figure 4.6 only shows the i^{th} column of a Multi-Mesh and the four fiber links between the router R_{i1} and block B_{i1}. All the routers have similar connections to the corresponding blocks.

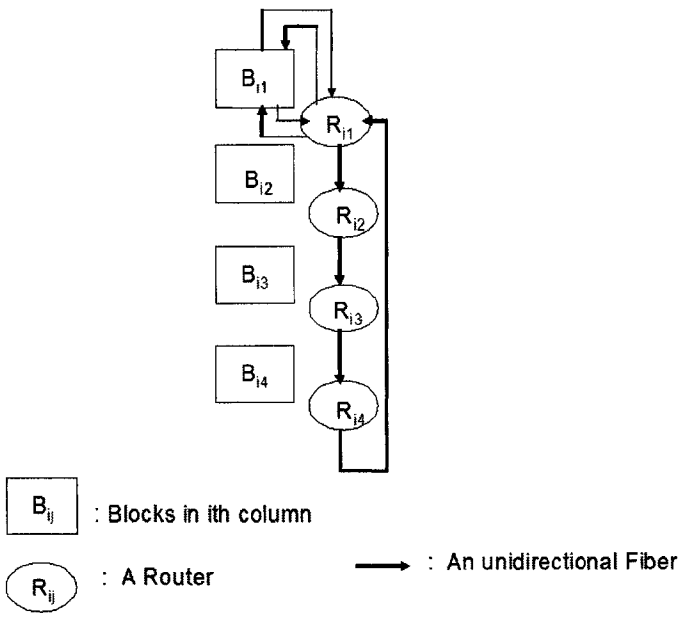

Figure 4.6: Connection between router R_{11} and block B_{11}

4.1.1 Physical topology using unidirectional links

The block diagram shown in Figure 4.7 is identical to that shown in Figure 4.3 except that the links have directions as shown.

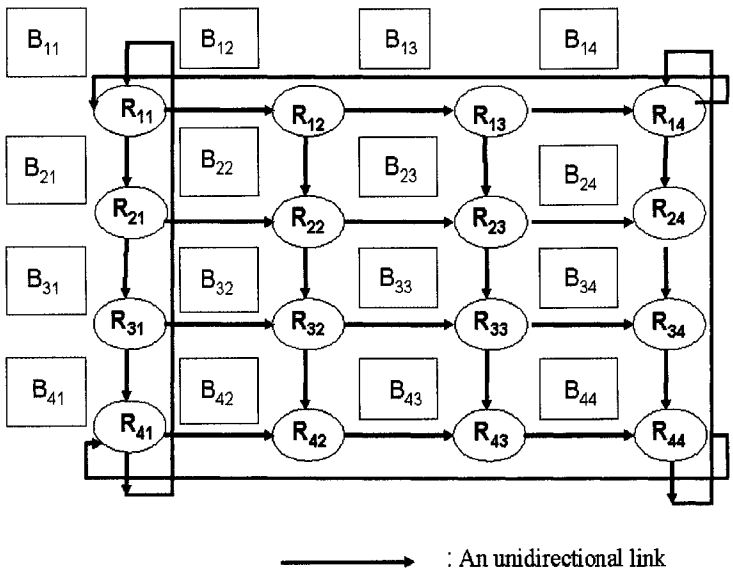

Figure 4.7: A MM network based on unidirectional links

We are discussing here only the implementation of the vertical inter-block connections since the horizontal inter-block connections may be achieved in exactly the same way.

4.1.2 Physical topology using bidirectional link

If we use bi-directional links, the only difference is that a link between router x and router y actually corresponds to a fiber from x to y and a fiber from y to x. As shown in Figure 4.8, we denote a link between x and y by $x \leftrightarrow y$.

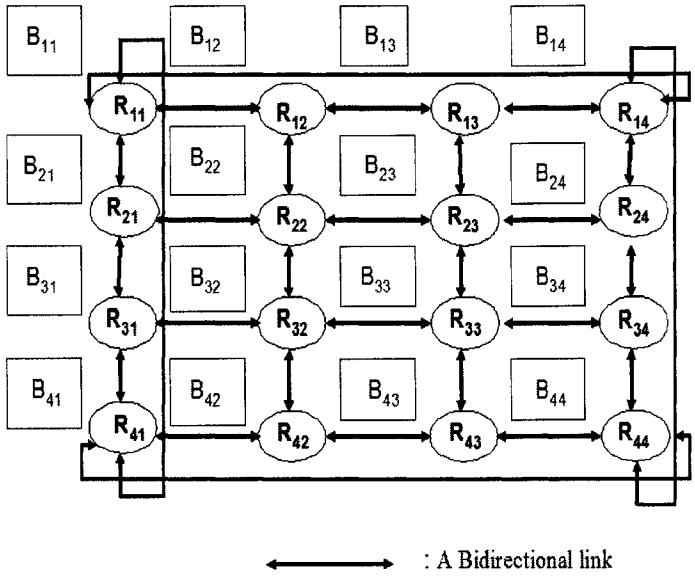

Figure 4.8: A MM network based on bidirectional links

4.2 Logical Topology for a Fault-Free Multi-Mesh

As mentioned earlier, our logical topology must have a directed edge for each inter-block connection. Here we only discuss the vertical inter-block links since the case for the horizontal inter-block links are identical. In a Multi-Mesh of order n, the boundary processors on the top (bottom) edge of block $B(\alpha, \beta)$, are connected to the boundary processors on the bottom (top) edge of block $B(*, \beta)$. In other words, processors $P(\alpha, \beta, 1, y)$ ($P(\alpha, \beta, n, y)$) are connected to processor $P(y, \beta, n, \alpha)$ ($P(y, \beta, 1, \alpha)$), for all y, $1 \leq y \leq n$, $y \neq \alpha$.

In our problem, we need two lightpaths from each block $B_{\alpha, \beta}$ to block $B_{y, \beta}$ - one for the connection from processor $P(\alpha, \beta, 1, y)$ to $P(y, \beta, n, \alpha)$ and one for the connection from

processor $P(\alpha, \beta, n, y)$ to $P(y, \beta, 1, \alpha)$ for all α, y, $1 \leq \alpha$, $y \leq n$. We now look at the ring consisting only of the routers in column number β and the fibers connecting them. We may view the $B_{\alpha, \beta}$, as the end node connected by the multiplexer collecting lightpaths from all processors on the top edge of the block to router $R_{\alpha, \beta}$. The set of lightpaths from the top edges of the blocks $B_{\alpha, \beta}$, $1 \leq \alpha \leq n$ define a completely connected ring. Similarly the set of lightpaths from the bottom edges of the blocks $B_{\alpha, \beta}$, $1 \leq \alpha \leq n$ define another completely connected ring. In summary our problem is to define complete connectivity for a unidirectional ring using a set of wavelengths say $\{\lambda_1, \lambda_{2, \ldots} \lambda_K\}$. This constitutes the set of connections from all the processors on the top edge of block in column β. Then we define an independent second set of complete connections simply by using another set of wavelengths $\{\lambda_{K+1}, \lambda_{K+2, \ldots} \lambda_{2K.}\}$. This second set constitutes the set of connections from all the processors on the bottom edge of block in column β. Research has already been done on wavelength assignment in bi-directional WDM rings [StBa99] and a recursive procedure for wavelength assignment for complete connectivity in bi-directional rings has been reported [EBC98].

4.2.1 Logical topology using unidirectional links

We now describe our process for assigning routes and wavelengths to each lightpath to define complete connectivity for a unidirectional ring. Due to the symmetric nature of our network, we have chosen a straight forward route for our lightpaths - we will use only the fibers connecting routers in column β when defining lightpaths from any block in column β to any other block in the same column. We will use the following algorithm to assign

wavelengths to each lightpath. This algorithm assumes that there is a unidirectional ring with N nodes, $1 < N < n$ which are assigned numbers 1, 2, ... , N with wavelengths already assigned to them for complete connectivity. The algorithm simply puts a new node ($Node_{N+1}$), in any desired position on the ring and assigns wavelengths for communication from every existing node to the new node and wavelengths for communication from the new node to every existing node. We assume that node N + 1 is placed after node i shown in Figure 4.9, in the network. We will use new wavelengths $\lambda_{N+1}^{1}, \lambda_{N+1}^{2}, \ldots, \lambda_{N+1}^{N}$ in our algorithm.

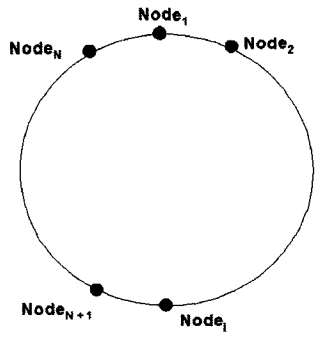

Figure 4.9: Inserting the (N+1)th node in a unidirectional ring

Algorithm Assign-wavelength

Step 1) repeat step 2 for all j, $1 \leq j \leq N$

Step 2) assign wavelength λ_{N+1}^{j} for communication from node j to node N + 1

Step 3) assign wavelength λ_{N+1}^{j} for communication from node N + 1 to node j.

To assign wavelengths for all nodes 1, 2, ... , n, we simply start with 2 nodes which require 1 wavelength for communication and then keep adding nodes 3, 4, ... n. The total number of wavelengths needed is $1 + 2 + \ldots + (n\text{-}1) = n(n\text{-}1)/2$.

4.2.2 Logical topology using bidirectional links

We will use the same route chosen in the previous section so that we will again use only the fibers connecting routers in column β when defining lightpaths from any block in column β to any other block in the same column. We already have an algorithm for assigning routes and wavelengths to each lightpath to define complete connectivity for a bidirectional ring [EBC98]. They also chose a shortest path routing and have described a recursive algorithm to determine the wavelengths needed for complete connectivity. We will use their algorithm. Since we need to define two lightpaths from each end node to every other end node, we will need $K = (n^2 - 1)/8$ wavelengths.

4.3 Robust Logical Topology for a Multi-Mesh

Faults in interconnection networks have been investigated for a long time [Ge98], [RaMu99b], [SRM02]. The standard approach in designing fault tolerant interconnection networks is that, in the case of faults, we have to determine a path edge (node) disjoint from the faulty edge (node). In other words, to avoid faults, in the standard approach, the message has to use a different routing algorithm where the message passes through a sequence of processors different from that used in the absence of faults.

We are proposing a scheme for tolerating faults in the optical path that we may include at relatively little cost. This scheme uses the protection path scheme, so that if there is a fault affecting a number of lightpaths, we can use alternate optical paths. In other words, even though the optical path used in sending a message does change, the diameter and the routing algorithm remains the same.

We now discuss how we may handle the case of faults in the logical topology. In our scheme we make the following assumptions:

- our physical topology uses bidirectional links,
- we do not have to deal with more than one fault at a time.

We will use path protection schemes [SRM02] that has been proposed recently for wide area optical networks. In a path protection scheme, when defining lightpaths, it is ensured that additional optical resources are included in the network so that every lightpath affected by any fault in the network may be rerouted to avoid the faulty element. In the absence of faults, *primary* paths are used for all communication [GeRa00]. When a single fault occurs, a number of lightpaths passing through the fault (resulting from a cut in the fiber, fault in the receiver or transmitter) will no longer be usable. Each of these failed lightpaths must be rerouted so that they use a backup path that does not use the faulty element affecting its primary path. To achieve this, we have to make sure that the spare capacity in the optical part of the network is sufficient to allow the creation of such backup lightpaths when needed. At the same time, for reasons of economy, the amount of additional resources needed to guarantee that backup paths may be created in all possible situations must be kept to a minimum. *Shared path protection* is used to

minimize the additional overhead needed to create the backup paths [RaMu99a]. In shared path protection, the following rules must be followed [RaMu99a]:

- the primary path and the backup path for a given lightpath must be edge-disjoint,
- two primary paths sharing a fiber must be assigned different wavelengths,
- two backup paths may share a fiber as well as have the same wavelength provided the corresponding primary paths are edge disjoint (since we assume single fault these two fiber-disjoint primary paths cannot fail at the same time).

We will now discuss how we may incorporate protection path scheme using the bi-directional optical networks discussed earlier in this chapter. In describing this scheme, we have to

- indicate the primary paths for each of the inter-block connections,
- indicate how we can define protection paths to handle every possible fault,
- calculate the cost of such a scheme.

Our scheme uses, for primary paths, the same paths we used in defining the logical topology of fault free networks using bi-directional links. We will use the same set of $K = (n^2 - 1)/8$ wavelengths used there.

We now consider the case of a single fault. In describing the approach we will use addition (+) or subtraction on rows and columns. It should be noted that there is a "wrap-around" so that row (or column) n is followed by row (or column) 1. We only discuss the failure of communication in the downward direction, say from router $R_{\alpha\beta}$ to router

$R_{(\alpha+1)\beta}$. This failure may be due to a failure in the fiber or due to a fault either in the router $R_{\alpha\beta}$ or $R_{(\alpha+1)\beta}$. The case of a failure of communication in the horizontal direction or in the upward direction is identical. We note that the fiber from router $R_{\alpha\beta}$ to router $R_{(\alpha+1)\beta}$ is used by a primary lightpath from block $B_{\delta\beta}$ to block $B_{(\delta+m)\beta}$ if and only if $m \leq n/2$ and $\delta \leq \alpha \leq \delta + m$. Our scheme for setting up backup lightpaths therefore only needs to consider the failed lightpaths that happen due to a fault in communication from router $R_{\alpha\beta}$ to router $R_{(\alpha+1)\beta}$. For our convenience, we group these primary lightpaths and assign labels to them as follows:

Group 1 consisting of the following lightpaths to block $B_{(\alpha+1)\beta}$

- lightpath $PL_{(\alpha+1)1}$ from block $B_{\alpha\beta}$
- lightpath $PL_{(\alpha+1)2}$ from block $B_{(\alpha-1)\beta}$
- ....
- lightpath $PL_{(\alpha+1)(n/2)}$ from block $B_{(\alpha-n/2+1)\beta}$

Group 2 consisting of the following lightpaths to block $B_{(\alpha+2)\beta}$

- lightpath $PL_{(\alpha+2)1}$ from block $B_{\alpha\beta}$
- lightpath $PL_{(\alpha+2)2}$ from block $B_{(\alpha-1)\beta}$
- ....
- lightpath $PL_{(\alpha+2)(n/2-1)}$ from block $B_{(\alpha-n/2+2)\beta}$

...

Group $n/2$ consisting of the following lightpath to block $B_{(\alpha+n/2)\beta}$

- lightpath $PL_{(\alpha+n/2)1}$ from block $B_{\alpha\beta}$

Example:

Figure 4.10 shows an example of a MM network of order 8 where a fault has occurred affecting communicating from router R_{43} to router R_{53}.

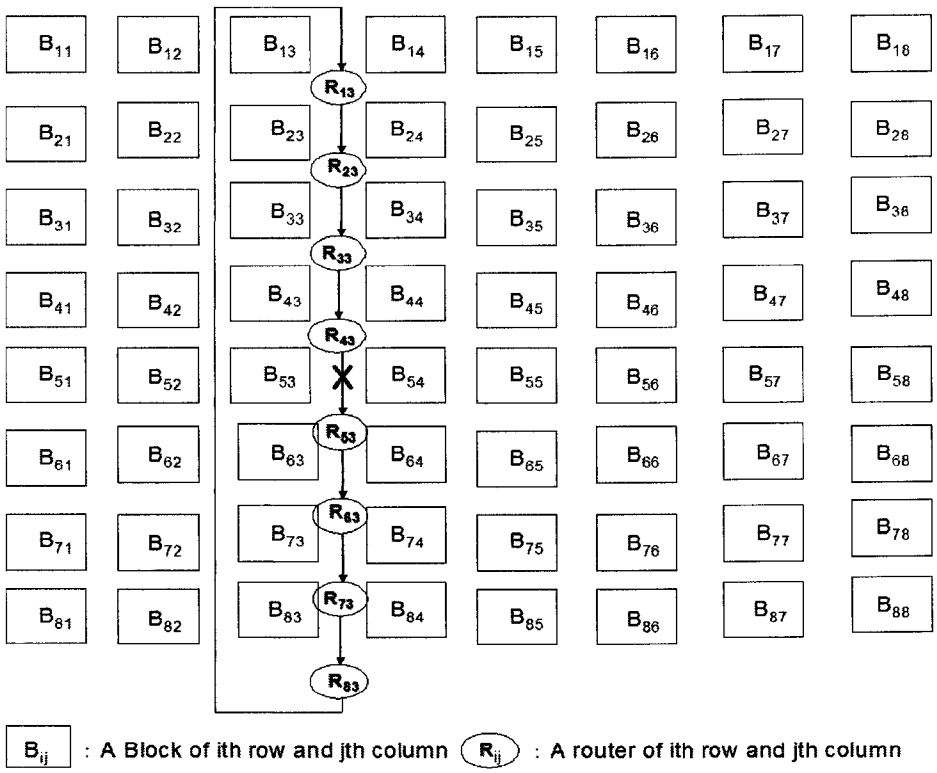

Figure 4.10: A faulty link in a multi-mesh of order 8

The primary lightpaths in the fibers of different groups are as shown in Table 4.1.

Table 4.1: Primary lightpaths in the fibers of different groups

Group number	Label of primary lightpath	Source node	Destination node
1	PL_{51}	B_{43}	B_{53}
	PL_{52}	B_{33}	B_{53}
	PL_{53}	B_{23}	B_{53}
	PL_{54}	B_{13}	B_{53}
2	PL_{61}	B_{43}	B_{63}
	PL_{62}	B_{33}	B_{63}
	PL_{63}	B_{23}	B_{63}
3	PL_{71}	B_{43}	B_{73}
	PL_{72}	B_{33}	B_{73}
4	PL_{81}	B_{43}	B_{83}

In our scheme for defining backup lightpaths, if n is even, we need additional $n/2$ wavelengths $\{\lambda_1, \lambda_2, \ldots \lambda_{n/2}\}$to implement our scheme. To define a backup lightpath for each primary lightpath affected by a fault anywhere in the network, we specify the route and the wavelength for each of the backup lightpaths that must replace an affected primary lightpath as follows:

a) To replace the primary lightpaths $PL_{(\alpha+i)\,k}$ in group i we use column $\beta + i$ to route the backup lightpath $BL_{(\alpha+i)\,k}$, $1 \leq i \leq n/2$, $1 \leq k \leq n/2 - i + 1$.

b) We assign wavelength λ_p to backup lightpath $BL_{(\alpha+i)\,k}$ where $p = (1\text{-}i) \oplus_{n/2} (k\text{-}1) + 1$

We note that when calculating p, we use "wrap-around" so that λ_1 is preceded by $\lambda_{n/2}$.

Example

Figure 4.11 shows the same multi-mesh shown in Figure 4.10 but, includes all the routers and the horizontal and vertical fibers. We have omitted the wraparound connections to simplify the diagram and we have not shown the connections from a block to the routers. Once again we are considering the fault be in the communication from R_{43} to R_{53}.

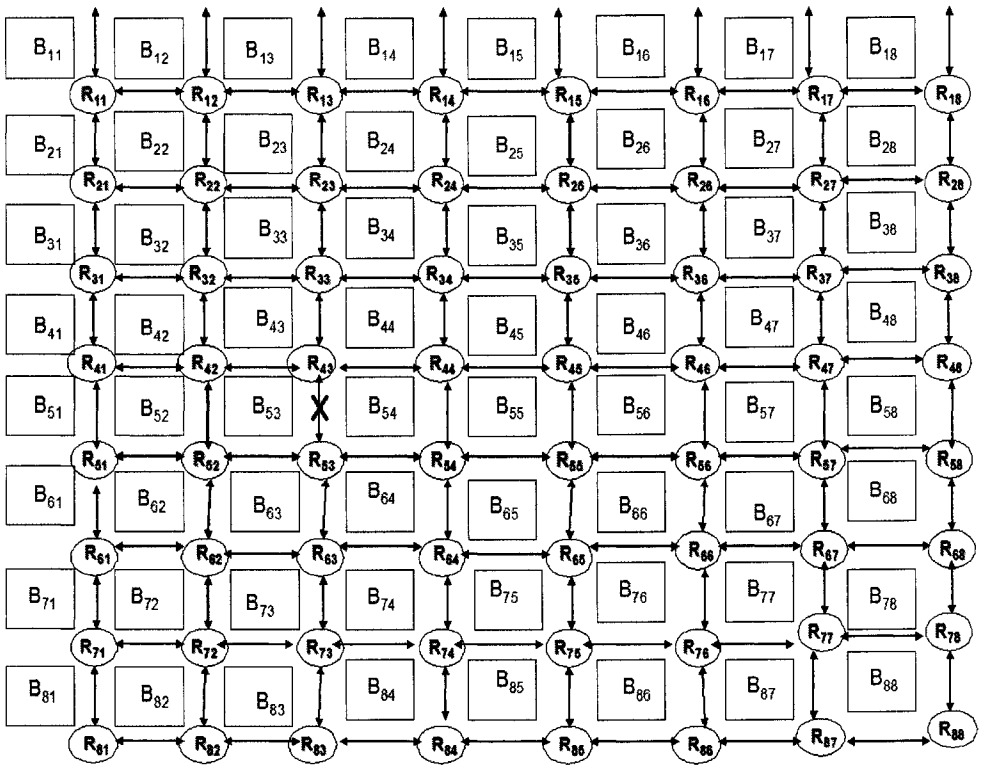

Figure 4.11: A faulty multi-mesh of order 8

Table 4.2 describes the details of the backup lightpaths. We will only explain one situation.

Table 4.2: The backup lightpaths

Group #	Primary Lightpath		Backup lightpath	
	originating from	Terminating at	Routers used	Wave-length
1	B_{43}	B_{53}	$R_{43} \rightarrow R_{44} \rightarrow R_{54} \rightarrow R_{53}$	λ_1
	B_{33}	B_{53}	$R_{33} \rightarrow R_{34} \rightarrow R_{44} \rightarrow R_{54} \rightarrow R_{53}$	λ_2
	B_{23}	B_{53}	$R_{23} \rightarrow R_{24} \rightarrow R_{34} \rightarrow R_{44} \rightarrow R_{54} \rightarrow R_{53}$	λ_3
	B_{13}	B_{53}	$R_{13} \rightarrow R_{14} \rightarrow R_{24} \rightarrow R_{34} \rightarrow R_{44} \rightarrow R_{54} \rightarrow R_{53}$	λ_4
2	B_{43}	B_{63}	$R_{43} \rightarrow R_{44} \rightarrow R_{45} \rightarrow R_{55} \rightarrow R_{65} \rightarrow R_{64} \rightarrow R_{63}$	λ_4
	B_{33}	B_{63}	$R_{33} \rightarrow R_{34} \rightarrow R_{35} \rightarrow R_{45} \rightarrow R_{55} \rightarrow R_{65} \rightarrow R_{64} \rightarrow R_{63}$	λ_1
	B_{23}	B_{63}	$R_{23} \rightarrow R_{24} \rightarrow R_{25} \rightarrow R_{35} \rightarrow R_{45} \rightarrow R_{55} \rightarrow R_{65} \rightarrow R_{64} \rightarrow R_{63}$	λ_2
3	B_{43}	B_{73}	$R_{43} \rightarrow R_{44} \rightarrow R_{45} \rightarrow R_{46} \rightarrow R_{56} \rightarrow R_{66} \rightarrow R_{76} \rightarrow R_{75} R_{74} \rightarrow R_{73}$	λ_3
	B_{33}	B_{73}	$R_{33} \rightarrow R_{34} \rightarrow R_{35} \rightarrow R_{36} \rightarrow R_{36} \rightarrow R_{56} \rightarrow R_{66} \rightarrow R_{76} \rightarrow R_{73}$	λ_4
4	B_{43}	B_{83}	$R_{43} \rightarrow R_{44} \rightarrow R_{45} \rightarrow R_{46} \rightarrow R_{47} \rightarrow R_{57} \rightarrow R_{67} \rightarrow R_{77} R_{56} \rightarrow R_{87} \rightarrow R_{86} \rightarrow R_{85} \rightarrow R_{84} \rightarrow R_{83}$	λ_2

Group 2 includes primary lightpath PL_{62} from block B_{33} to B_{63}. The primary path passes through the faulty link from router R_{43} to R_{53}. The corresponding backup lightpath BL_{62} will use the route $B_{33} \rightarrow R_{33} \rightarrow R_{34} \rightarrow R_{35} \rightarrow R_{45} \rightarrow R_{55} \rightarrow R_{65} \rightarrow R_{64} \rightarrow R_{63} \rightarrow B_{63}$. The wavelength of BL_{62} will be λ_p where $p = (1 - i) + (k - 1)$. Our pool of additional wavelengths consists of wavelengths $[\lambda_1, \lambda_2, \lambda_3, \lambda_4]$. Since the group number is 2, $i = 2.\ 1 - 2 = -1$ which corresponds to λ_4. Here $k = 2$ so that $k-1 = 1$. The wavelength immediately after λ_4 is λ_1.

In this chapter we have considered the implementation of the inter-block connections in a Multi-Mesh using optical links. We have considered the cases of using uni-directional as well as bi-directional links. We have proposed a new scheme for handling faults affecting the lightpaths where the routing algorithm is unaffected by single optical faults.

Chapter 5

Conclusions and Future Directions

5.1 Summary of Work Done

In this thesis we have proposed a new network topology called 3D Multi-Mesh (3D MM) for multiprocessor architecture which is an extension of a recently proposed architecture named Multi-Mesh. The main results of our investigations are as follows:

1) We have proposed a new architecture that uses the 3-dimensional mesh as its building block rather than a 2-dimensional mesh as done in the Multi-Mesh [DDS99]. We have shown that our architecture has better topological properties compared to the Multi-Mesh architecture and that a number of algorithms can be efficiently mapped on the 3D MM network.

2) We have explored a number of possible approaches for implementing the Multi-Mesh architecture using opto-electronic technologies. There are two novel features of our approach:
    a. We have shown that WDM wavelength-routed networks may be used to realize some of the links.
    b. We have shown that single faults may be handled easily without increasing the number of optical paths used.

5.2 Suggestions for Future Work

- We have presented the fundamental algorithms; still there are a number of basic algorithms like matrix multiplication, matrix transposes, and sorting etc. that can be efficiently mapped on 3D MM network.

- We have proposed a new network topology consists of n^6 processors, in real life there may be a situation where the number of processors may be less than or greater than n^6 of processors, in order to accommodate any number of processors incomplete 3D MM can be defined in the same way of incomplete Multi-Mesh.

- In our optical implementation of Multi-Mesh, we have taken care of one fiber link failure; it can be improved to two or more.

- We have proposed possible approaches of implementing Multi-Mesh using optical technology. Our results may be extended to define 3D Multi-Mesh using optical technology.

5.3 Concluding Remarks

We have proposed a new architecture for interconnection networks and have shown that the proposed network has significantly better topological properties (e.g., diameter, node degree) compared to other mesh-based network, specially the Multi-Mesh network. We have established the fundamental algorithm for summation/average/maximum/minimum and point-to-point communication and shown that this network outperforms the Multi-Mesh network. Our optical implementation has the following novel features:

- we have used wavelength routed WDM networks which has lower power requirements compared to passive star coupler based designs used in other optical implementations

- we have shown that protection schemes may be used in this network with relatively little cost.

Appendix A

Path Computations for Different Source and Destination Pairs

Assumptions:

1. $\beta 2 > x1$
2. $\beta 1 > x2$
3. $\gamma 2 > y1$
4. $\gamma 1 > y2$
5. $\alpha 2 > z1$
6. $\alpha 1 > z2$

In order to path calculation in diameter of 3D MM we have assumed the above 6 cases. As there are 6 assumptions there can be all together 64 cases as shown in the table.

Table: 64 possible cases of source and destination

Cases	$\beta 2 > x1$	$\beta 1 > x2$	$\gamma 2 > y1$	$\gamma 1 > y2$	$\alpha 2 > z1$	$\alpha 1 > z2$
1	T	T	T	T	T	T
2	T	T	T	T	T	F
3	T	T	T	T	F	T
4	T	T	T	T	F	F
5	T	T	T	F	T	T
6	T	T	T	F	T	F
7	T	T	T	F	F	T
8	T	T	T	F	F	F
9	T	T	F	T	T	T
10	T	T	F	T	T	F
11	T	T	F	T	F	T
12	T	T	F	T	F	F
13	T	T	F	F	T	T
14	T	T	F	F	T	F
15	T	T	F	F	F	T
16	T	T	F	F	F	F
17	T	F	T	T	T	T

18	T	F	T	T	T	F
19	T	F	T	T	F	T
20	T	F	T	T	F	F
21	T	F	T	F	T	T
22	T	F	T	F	T	F
23	T	F	T	F	F	T
24	T	F	T	F	F	F
25	T	F	F	T	T	T
26	T	F	F	T	T	F
27	T	F	F	T	F	T
28	T	F	F	T	F	F
29	T	F	F	F	T	T
30	T	F	F	F	T	F
31	T	F	F	F	F	T
32	T	F	F	F	F	F
33	F	T	T	T	T	T
34	F	T	T	T	T	F
35	F	T	T	T	F	T
36	F	T	T	T	F	F
37	F	T	T	F	T	T
38	F	T	T	F	T	F
39	F	T	T	F	F	T
40	F	T	T	F	F	F
41	F	T	F	T	T	T
42	F	T	F	T	T	F
43	F	T	F	T	F	T
44	F	T	F	T	F	F
45	F	T	F	F	T	T
46	F	T	F	F	T	F
47	F	T	F	F	F	T
48	F	T	F	F	F	F
49	F	F	T	T	T	T
50	F	F	T	T	T	F
51	F	F	T	T	F	T
52	F	F	T	T	F	F
53	F	F	T	F	T	T
54	F	F	T	F	T	F
55	F	F	T	F	F	T
56	F	F	T	F	F	F
57	F	F	F	T	T	T
58	F	F	F	T	T	F
59	F	F	F	T	F	T
60	F	F	F	T	F	F
61	F	F	F	F	T	T
62	F	F	F	F	T	F
63	F	F	F	F	F	T
64	F	F	F	F	F	F

We have taken care of the 64 possible cases; due to lack of space here we're showing some interesting cases. We first showed the block level and then in the processor level.

(Case 1) (All the 6 assumptions are true)

PT1: B ($\alpha1, \beta1, \gamma1$) → B ($\alpha2, \beta1, \gamma1$) → B ($\alpha2, \beta2, \gamma1$) → B ($\alpha2, \beta2, \gamma2$)

P ($\alpha1, \beta1, \gamma1$, x1, y1, z1) → P ($\alpha1, \beta1, \gamma1$, 1, y1, $\alpha2$) → P ($\alpha2, \beta1, \gamma1, n$, y1, $\alpha1$)

→ P ($\alpha2, \beta1, \gamma1, \beta2$, 1, $\alpha1$) → P ($\alpha2, \beta2, \gamma1, \beta1, n, \alpha1$) → P ($\alpha2, \beta2, \gamma1, \beta1, \gamma2, n$)

→ P ($\alpha2, \beta2, \gamma2, \beta1, \gamma1$, 1) → P ($\alpha2, \beta2, \gamma2$, x2, y2, z2)

Path Length:

(x1 - 1) + ($\alpha2$ – z1) + 1 + (n- $\beta2$) + (y1 – 1) + 1 + (n - $\gamma2$) + (n - $\alpha1$) + 1 + ($\beta1$- x2) + ($\gamma1$ – y2) + (z2 - 1)

= x1 - 1 + $\alpha2$ – z1 + 1 + n- $\beta2$ + y1– 1 + 1 + n - $\gamma2$ + n - $\alpha1$ + 1 + $\beta1$- x2 + $\gamma1$– y2 + z2 -1

= 3*n* + x1 + y1 - z1 - $\alpha1$ + $\beta1$ + $\gamma1$ - x2 - y2 + z2 + $\alpha2$ - $\beta2$ - $\gamma2$

PT2: B ($\alpha1, \beta1, \gamma1$) → B ($\alpha1, \beta1, \gamma2$) → B ($\alpha1, \beta2, \gamma2$) → B ($\alpha2, \beta2, \gamma2$)

P($\alpha1, \beta1, \gamma1$, x1, y1, z1) → P($\alpha1, \beta1, \gamma1$, x1, $\gamma2$, 1) → P($\alpha1, \beta1, \gamma2$, x1, $\gamma1, n$)

→ P($\alpha1, \beta1, \gamma2, \beta2, n, n$) → P($\alpha1, \beta2, \gamma2, \beta1$, 1, n) → P($\alpha1, \beta2, \gamma2, n$, 1, $\alpha2$)

→ P($\alpha2, \beta2, \gamma2$, 1 ,1, $\alpha1$) → P($\alpha2, \beta2, \gamma2$, x2, y2, z2)

Path Length:

= ($\gamma2$-y1) + (z1-1)+1+($\beta2$-x1) +(n-$\gamma1$)+1+(n-$\beta1$) + (n-$\alpha2$) + 1 + (x2-1) + (y2–1) + ($\alpha1$-z2)

= $\gamma2$ - y1 + z1 - 1+ 1 + $\beta2$ - x1 + n - $\gamma1$ + 1 + n - $\beta1$ + n - $\alpha2$ + 1 + x2 - 1 + y2 – 1+ $\alpha1$-z2

= 3*n* – x1 – y1 + z1 + $\alpha1$ - $\beta1$ - $\gamma1$ + x2 + y2 – z2 - $\alpha2$ + $\beta2$ + $\gamma2$

PT1 + PT2 :

PT1: 3*n* + x1 + y1 - z1 - $\alpha1$ + $\beta1$ + $\gamma1$ - x2 - y2 + z2 + $\alpha2$ - $\beta2$ - $\gamma2$

PT2: 3*n* – x1 – y1 + z1 + $\alpha1$ - $\beta1$ - $\gamma1$ + x2 + y2 – z2 - $\alpha2$ + $\beta2$ + $\gamma2$

= 3*n*

(Case 2:) (Assumption 6 false)

PT1: B ($\alpha1, \beta1, \gamma1$) → B ($\alpha2, \beta1, \gamma1$) → B ($\alpha2, \beta2, \gamma1$) → B ($\alpha2, \beta2, \gamma2$)

P ($\alpha1, \beta1, \gamma1$, x1, y1, z1) → P ($\alpha1, \beta1, \gamma1$, 1, y1, $\alpha2$) → P ($\alpha2, \beta1, \gamma1, n$, y1, $\alpha1$)

→ P ($\alpha2, \beta1, \gamma1, \beta2$, 1, $\alpha1$) → P ($\alpha2, \beta2, \gamma1, \beta1, n, \alpha1$) → P ($\alpha2, \beta2, \gamma1, \beta1, \gamma2$, 1)

→ P ($\alpha2, \beta2, \gamma2, \beta1, \gamma1, n$) → P ($\alpha2, \beta2, \gamma2$, x2, y2, z2)

Path Length:

(x1 - 1) + ($\alpha2$ – z1) + 1 + (n- $\beta2$) +(y1 – 1) + 1 + (n - $\gamma2$) + ($\alpha1$ - 1) + 1 + ($\beta1$- x2) +

($\gamma1$ – y2) + (n - z2)

= x1 - 1 + $\alpha2$ – z1 + 1 + n- $\beta2$ + y1– 1 + 1 + n - $\gamma2$ + n + $\alpha1$ + 1 + $\beta1$- x2 + $\gamma1$– y2 - z2 -1

= 3n + x1 + y1 - z1 + $\alpha1$ + $\beta1$ + $\gamma1$ - x2 - y2 - z2 + $\alpha2$ - $\beta2$ - $\gamma2$

PT2: B ($\alpha1, \beta1, \gamma1$) → B ($\alpha1, \beta1, \gamma2$) → B ($\alpha1, \beta2, \gamma2$) → B ($\alpha2, \beta2, \gamma2$)

P ($\alpha1, \beta1, \gamma1$, x1, y1, z1) → P ($\alpha1, \beta1, \gamma1$, x1, $\gamma2$, 1) → P ($\alpha1, \beta1, \gamma2$, x1, $\gamma1, n$)

→ P ($\alpha1, \beta1, \gamma2, \beta2, n, n$) → P ($\alpha1, \beta2, \gamma2, \beta1$, 1, n) → P ($\alpha1, \beta2, \gamma2, n$, 1, $\alpha2$)

→ P ($\alpha2, \beta2, \gamma2$, 1, 1, $\alpha1$) → P ($\alpha2, \beta2, \gamma2$, x2, y2, z2)

Path Length:

= ($\gamma2$-y1) + (z1-1) +1+ ($\beta2$-x1) + (n-$\gamma1$) +1+ (n-$\beta1$) + (n-$\alpha2$) +1+ (x2-1)+ (y2–1)+(z2-$\alpha1$)

= $\gamma2$ -y1 +z1 - 1+ 1 + $\beta2$ - x1 + n - $\gamma1$ + 1 + n - $\beta1$ + n - $\alpha2$ + 1 + x2 - 1 + y2 – 1+ $\alpha1$ + z2

= 3n – x1 – y1 + z1 + $\alpha1$ - $\beta1$ - $\gamma1$ + x2 + y2 – z2 - $\alpha2$ + $\beta2$ + $\gamma2$

PT1 + PT2 :

PT1: 3n + x1 + y1 - z1 + $\alpha1$ + $\beta1$ + $\gamma1$ - x2 - y2 - z2 + $\alpha2$ - $\beta2$ - $\gamma2$

PT2: 3n – x1 – y1 + z1 - $\alpha1$ - $\beta1$ - $\gamma1$ + x2 + y2 + z2 - $\alpha2$ + $\beta2$ + $\gamma2$

= 3n

(Case 3) (Assumptions 4 & 5 & 6 False)

PT1: B ($\alpha1, \beta1, \gamma1$) $\rightarrow$ B ($\alpha2, \beta1, \gamma1$) $\rightarrow$ B ($\alpha2, \beta2, \gamma1$) $\rightarrow$ B ($\alpha2, \beta2, \gamma2$)

P($\alpha1, \beta1, \gamma1$, x1, y1, z1) $\rightarrow$ P($\alpha1, \beta1, \gamma1$, 1, y1, $\alpha2$) $\rightarrow$ P($\alpha2, \beta1, \gamma1$, *n*, y1, $\alpha1$)

$\rightarrow$ P($\alpha2, \beta1, \gamma1, \beta2$, 1, $\alpha1$) $\rightarrow$ P($\alpha2, \beta2, \gamma1, \beta1$, *n* , $\alpha1$) $\rightarrow$ P($\alpha2, \beta2, \gamma1, \beta1, \gamma2$, 1)

$\rightarrow$ P($\alpha2, \beta2, \gamma2, \beta1, \gamma1$, *n*) $\rightarrow$ P($\alpha2, \beta2, \gamma2$, x2, y2, z2)

Path Length:

(x1-1) + (z1-$\alpha2$) + 1 + (*n*-$\beta2$) +(y1–1)+1+ (*n* -$\gamma2$)+ ($\alpha1$-1) +1+ ($\beta1$- x2)+(y2 -$\gamma1$)+(*n*-z2)

= x1-1 - $\alpha2$ + z1 + 1 + *n*- $\beta2$ + y1– 1 + 1 + *n* - $\gamma2$ + $\alpha1$ – 1 + 1 + $\beta1$- x2 - $\gamma1$+ y2 + *n* - z2

= 3*n* + x1 + y1 + z1 + $\alpha1$ + $\beta1$ - $\gamma1$ - x2 + y2 - z2 - $\alpha2$ - $\beta2$ - $\gamma2$

PT2: $\alpha1\beta1\gamma1 \rightarrow \alpha1\beta1\gamma2 \rightarrow \alpha1\beta2\gamma2 \rightarrow \alpha2\beta2\gamma2$

P($\alpha1, \beta1, \gamma1$, x1, y1, z1) $\rightarrow$ P($\alpha1, \beta1, \gamma1$, x1, $\gamma2$, *n*) $\rightarrow$ P($\alpha1, \beta1, \gamma2$, x1, $\gamma1$, 1)

$\rightarrow$ P($\alpha1, \beta1, \gamma2, \beta2$, 1, 1) $\rightarrow$ P($\alpha1, \beta2, \gamma2, \beta1$, *n*, 1) $\rightarrow$ P($\alpha1, \beta2, \gamma2$, *n*, *n*, $\alpha2$)

$\rightarrow$ P($\alpha2, \beta2, \gamma2$, 1 ,*n*, $\alpha1$) $\rightarrow$ P($\alpha2, \beta2, \gamma2$, x2, y2, z2)

Path Length:

= ($\gamma2$-y1) + (*n*-z1) + 1+ ($\beta2$-x1)+($\gamma1$–1)+1+(*n*-$\beta1$)+ ($\alpha2$–1) + 1 +(x2-1)+(*n*-y2)+ (z2 - $\alpha1$)

= $\gamma2$- y1 - z1 - 1+ 1 + $\beta2$ - x1 + *n* + $\gamma1$ - 1 + *n* - $\beta1$ + *n* + $\alpha2$ + 1 + x2 - 1 + *n* - y2 - $\alpha1$+z2

= 3*n* – x1 – y1 - z1 - $\alpha1$ - $\beta1$ + $\gamma1$ + x2 - y2 + z2 + $\alpha2$ + $\beta2$ + $\gamma2$

PT1 + PT2 :

PT1: 3*n* + x1 + y1 + z1 + $\alpha1$ + $\beta1$ - $\gamma1$ - x2 + y2 - z2 - $\alpha2$ - $\beta2$ - $\gamma2$

PT2: 3*n* – x1 – y1 - z1 - $\alpha1$ - $\beta1$ + $\gamma1$ + x2 - y2 + z2 + $\alpha2$ + $\beta2$ + $\gamma2$

= 3*n*

(Case 4) (Assumptions 3 & 4 & 5 & 6 are false)

PT1: B ($\alpha1, \beta1, \gamma1$) → B ($\alpha2, \beta1, \gamma1$) → B ($\alpha2, \beta2, \gamma1$) → B ($\alpha2, \beta2, \gamma2$)

$P(\alpha1, \beta1, \gamma1, x1, y1, z1) \rightarrow P(\alpha1, \beta1, \gamma1, 1, y1, \alpha2) \rightarrow P(\alpha2, \beta1, \gamma1, n, y1, \alpha1)$

$\rightarrow P(\alpha2, \beta1, \gamma1, \beta2, 1, \alpha1) \rightarrow P(\alpha2, \beta2, \gamma1, \beta1, n, \alpha1) \rightarrow P(\alpha2, \beta2, \gamma1, \beta1, \gamma2, 1)$

$\rightarrow P(\alpha2, \beta2, \gamma2, \beta1, \gamma1, n) \rightarrow P(\alpha2, \beta2, \gamma2, x2, y2, z2)$

Path Length:

$(x1-1)+(z1-\alpha2)+1+(n-\beta2)+(n-y1)+1+(\gamma2-1)+(\alpha1-1)+1+(\beta1-x2)+(y2-\gamma1)+(n-z2)$

$= x1 - 1 + z1 - \alpha2 + 1 + n - \beta2 + n - y1 + 1 + \gamma2 - 1 + n + \alpha1 + 1 + \beta1 - x2 - \gamma1 + y2 - z2 - 1$

$= \mathbf{3n + x1 - y1 + z1 + \alpha1 + \beta1 - \gamma1 - x2 + y2 - z2 - \alpha2 - \beta2 + \gamma2}$

PT2: B ($\alpha1, \beta1, \gamma1$) → B ($\alpha1, \beta1, \gamma2$) → B ($\alpha1, \beta2, \gamma2$) → B ($\alpha2, \beta2, \gamma2$)

$P(\alpha1, \beta1, \gamma1, x1, y1, z1) \rightarrow P(\alpha1, \beta1, \gamma1, x1, \gamma2, n) \rightarrow P(\alpha1, \beta1, \gamma2, x1, \gamma1, 1)$

$\rightarrow P(\alpha1, \beta1, \gamma2, \beta2, 1, 1) \rightarrow P(\alpha1, \beta2, \gamma2, \beta1, n, 1) \rightarrow P(\alpha1, \beta2, \gamma2, n, n, \alpha2)$

$\rightarrow P(\alpha2, \beta2, \gamma2, 1, n, \alpha1) \rightarrow P(\alpha2, \beta2, \gamma2, x2, y2, z2)$

Path Length:

$= (y1-\gamma2)+(n-z1)+1+(\beta2-x1)+(\gamma1-1)+1+(n-\beta1)+(\alpha2-1)+1+(x2-1)+(n-y2)+(z2-\alpha1)$

$= y1 - \gamma2 - z1 - 1 + 1 + \beta2 - x1 + n + \gamma1 + 1 + n - \beta1 + n + \alpha2 + 1 + x2 - 1 - y2 - 1 - \alpha1 + z2$

$= \mathbf{3n - x1 + y1 - z1 - \alpha1 - \beta1 + \gamma1 + x2 - y2 + z2 + \alpha2 + \beta2 - \gamma2}$

PT1 + PT2 :

PT1: $\mathbf{3n + x1 - y1 + z1 + \alpha1 + \beta1 - \gamma1 - x2 + y2 - z2 - \alpha2 - \beta2 + \gamma2}$

PT2: $\mathbf{3n - x1 + y1 + z1 - \alpha1 - \beta1 + \gamma1 + x2 - y2 + z2 + \alpha2 + \beta2 - \gamma2}$

$= \mathbf{3n}$

(Case 5) (Assumptions 2 & 3 & 4 & 5 & 6 are false)

PT1: B ($\alpha1, \beta1, \gamma1$) $\rightarrow$ B ($\alpha2, \beta1, \gamma1$) $\rightarrow$ B ($\alpha2, \beta2, \gamma1$) $\rightarrow$ B ($\alpha2, \beta2, \gamma2$)

$P(\alpha1, \beta1, \gamma1, x1, y1, z1) \rightarrow P(\alpha1, \beta1, \gamma1, 1, y1, \alpha2) \rightarrow P(\alpha2, \beta1, \gamma1, n, y1, \alpha1)$

$\rightarrow P(\alpha2, \beta1, \gamma1, \beta2, n, \alpha1) \rightarrow P(\alpha2, \beta2, \gamma1, \beta1, 1, \alpha1) \rightarrow P(\alpha2, \beta2, \gamma1, \beta1, \gamma2, 1)$

$\rightarrow P(\alpha2, \beta2, \gamma2, \beta1, \gamma1, n) \rightarrow P(\alpha2, \beta2, \gamma2, x2, y2, z2)$

Path Length:

$(x1-1)+(z1-\alpha2)+1+(n-\beta2)+(n-y1)+1+(\gamma2-1)+(\alpha1-1)+1+(x2-\beta1)+(y2-\gamma1)+(n-z2)$

$= x1 - 1 - \alpha2 + z1 + 1 + n - \beta2 - y1 - 1 + 1 + n + \gamma2 + \alpha1 + 1 - \beta1 + x2 - \gamma1 + y2 - z2 - 1$

$= \mathbf{3}n + \mathbf{x1 - y1 + z1 + \alpha1 - \beta1 - \gamma1 + x2 + y2 - z2 - \alpha2 - \beta2 + \gamma2}$

PT2: B ($\alpha1, \beta1, \gamma1$) $\rightarrow$ B ($\alpha1, \beta1, \gamma2$) $\rightarrow$ B ($\alpha1, \beta2, \gamma2$) $\rightarrow$ B ($\alpha2, \beta2, \gamma2$)

$P(\alpha1, \beta1, \gamma1, x1, y1, z1) \rightarrow P(\alpha1, \beta1, \gamma1, x1, \gamma2, n) \rightarrow P(\alpha1, \beta1, \gamma2, x1, \gamma1, 1)$

$\rightarrow P(\alpha1, \beta1, \gamma2, \beta2, 1, 1) \rightarrow P(\alpha1, \beta2, \gamma2, \beta1, n, 1) \rightarrow P(\alpha1, \beta2, \gamma2, 1, n, \alpha2)$

$\rightarrow P(\alpha2, \beta2, \gamma2, n, n, \alpha1) \rightarrow P(\alpha2, \beta2, \gamma2, x2, y2, z2)$

Path Length:

$= (y1-\gamma2) + (n - z1) + 1 + (\beta2-x1) + (\gamma1-1)+1+(\beta1-1)+(\alpha2-1)+1+(n-x2)+(n-y2)+(z2-\alpha1)$

$= y1-\gamma2 - z1 - 1 + 1 + \beta2 - x1 + n + \gamma1 + 1 + n + \beta1 + n + \alpha2 + 1 - x2 - 1 - y2 - 1 - \alpha1 + z2$

$= \mathbf{3}n - \mathbf{x1 + y1 - z1 - \alpha1 + \beta1 + \gamma1 - x2 - y2 + z2 + \alpha2 + \beta2 - \gamma2}$

PT1 + PT2 :

PT1: $\mathbf{3}n + \mathbf{x1 - y1 + z1 + \alpha1 - \beta1 - \gamma1 + x2 + y2 - z2 - \alpha2 - \beta2 + \gamma2}$

PT2: $\mathbf{3}n - \mathbf{x1 + y1 - z1 - \alpha1 + \beta1 + \gamma1 - x2 - y2 + z2 + \alpha2 + \beta2 - \gamma2}$

$= \mathbf{3}n$

(Case 6) (All the assumption 1 & 2 & 3 & 4 & 5 & 6 are false)

PT1: B ($\alpha1, \beta1, \gamma1$) → B ($\alpha2, \beta1, \gamma1$) → B ($\alpha2, \beta2, \gamma1$) → B ($\alpha2, \beta2, \gamma2$)

P($\alpha1, \beta1, \gamma1$, x1, y1, z1) → P($\alpha1, \beta1, \gamma1, n$, y1, $\alpha2$) → P($\alpha2, \beta1, \gamma1$, 1, y1, $\alpha1$)

→ P($\alpha2, \beta1, \gamma1, \beta2, n, \alpha1$) → P($\alpha2, \beta2, \gamma1, \beta1$, 1 , $\alpha1$) → P($\alpha2, \beta2, \gamma1, \beta1, \gamma2$, 1)

→ P($\alpha2, \beta2, \gamma2, \beta1, \gamma1, n$) → P($\alpha2, \beta2, \gamma2$, x2, y2, z2)

Path Length:

$(n$ - x1)+(z1-$\alpha2$) +1+($\beta2$ – 1)+(n-y1)+1+($\gamma2$-1)+($\alpha1$-1) +1+(x2 - $\beta1$) + (y2 -$\gamma1$)+(n - z2)

= - x1 -1 - $\alpha2$ + z1 + 1 + n + $\beta2$ - y1– 1 + 1 + n + $\gamma2$ + $\alpha1$ + 1 - $\beta1$+ x2 - $\gamma1$+ y2 - z2 -1

= 3*n* - x1 - y1 + z1 + $\alpha1$ - $\beta1$ - $\gamma1$ + x2 + y2 - z2 - $\alpha2$ + $\beta2$ + $\gamma2$

PT2: B ($\alpha1, \beta1, \gamma1$) → B ($\alpha1, \beta1, \gamma2$) → B ($\alpha1, \beta2, \gamma2$) → B ($\alpha2, \beta2, \gamma2$)

P($\alpha1, \beta1, \gamma1$, x1, y1, z1) → P($\alpha1, \beta1, \gamma1$, x1, $\gamma2, n$) → P($\alpha1, \beta1, \gamma2$, x1, $\gamma1$, 1)

→ P($\alpha1, \beta1, \gamma2, \beta2$, 1 , 1) → P($\alpha1, \beta2, \gamma2, \beta1, n$, 1) → P($\alpha1, \beta2, \gamma2$, 1, n, $\alpha2$)

→ P($\alpha2, \beta2, \gamma2, n, n$, $\alpha1$) → P($\alpha2, \beta2, \gamma2$, x2, y2, z2)

Path Length:

= (y1-$\gamma2$)+(n-z1) +1+ (x1-$\beta2$)+($\gamma1$–1)+1+($\beta1$-1)+($\alpha2$-1)+1+ (n- x2) + (n - y2) + (z2 - $\alpha1$)

= y1-$\gamma2$ - z1 - 1+ 1 - $\beta2$ + x1 + n + $\gamma1$ + 1 + n + $\beta1$ + n + $\alpha2$ + 1 - x2 - 1 - y2 – 1- $\alpha1$+z2

= 3*n* + x1 + y1 - z1 - $\alpha1$ + $\beta1$ + $\gamma1$ - x2 - y2 + z2 + $\alpha2$ - $\beta2$ - $\gamma2$

PT1 + PT2 :

PT1: 3*n* - x1 - y1 + z1 + $\alpha1$ - $\beta1$ - $\gamma1$ + x2 + y2 - z2 - $\alpha2$ + $\beta2$ +$\gamma2$

PT2: 3*n* + x1 + y1 - z1 - $\alpha1$ + $\beta1$ + $\gamma1$ - x2 - y2 + z2 + $\alpha2$ - $\beta2$ - $\gamma2$

= 3*n*

Appendix B

Glossary of Important Terms

Boundary Processor: The processors on the sides of the block (but not on corners) are characterized by x and y values such that exactly one of these coordinates are 1 or *n*.

Connectivity: The minimum number of arcs that have to be removed from the network to cut the network into two disconnected networks.

Corner Processor: The processors situated in the corner of a block meaning that all of the coordinate values are exactly 1 or *n*.

Cost: Total number of communication links of a network.

Diameter: Diameter of a graph is the maximum of the shortest distance (hops) between two nodes.

Dynamic Interconnection Network: Connections among the processors can be changed; the processors are not directly wired.

Fault Tolerant Optical Network: A network that is capable of working even in the presence of faults. In some cases WDM optical network provides alternate paths to avoid faults.

Free Space: Instead of optical fiber free space optical interconnection uses air space for optical signal propagation

Inter-Block Links: The links that connect the processor of different blocks

Interconnection Network: Interconnection network connects different processors in a multi-processor system.

Internal Processor: the processors in a block having all the connections (neighbors) within the bock.

Intra-Block Link: Processors within a block are connected by intra-block link.

Lightpath: The all-optical path through which the information flows in a wavelength-routed optical network. A lightpath may be composed of a single wavelength or it may consist of multimode of wavelengths.

Logical Topology: Is a graph that is obtained from the physical topology by assigning the lightpaths between the nodes. The nodes of the graph are the end-nodes of the physical topology and two nodes connected by a directed edge if there is a lightpath between them.

Multi-Hop Network: A network in which a packet may hop through zero or more intermediate nodes before it reaches its final destination.

Multiplexer/Demultiplexer: Optical multiplexers are used to combine several independent signals at different wavelength into one fiber. A demultiplexer works exactly the opposite way that is splitting the signals at different wavelengths.

Multiprocessor Architecture: A system consisting of more than one processing units where processors work simultaneously to solve a given problem.

Neighbors: Processors within the block that are directly connected are called neighbors.

Network Size: Total number of nodes in a network

Node Degree: Total number of incoming and outgoing links of a node.

Optical Communication: Data communication in a network where data is transmitted through optical fiber.

Optical Couplers: Coupler is a general term that covers all the devices that combine beams of light into or split into beams of light out of a fiber.

Optical Fiber: Optical fiber is a medium of data transmission where data is transmitted in the form of light wave. Optical fiber is a thin filament of glass, which acts as a wave-guide

Optical Router: In an optical network, a router is a device that is connected to a number of fibers, some carrying incoming optical signals to the router and the others carrying outgoing optical signals. A router determines how the incoming signals will be directed to the outgoing fibers.

Passive Star Coupler: The passive star coupler is a "broadcast" device, where an optical signal transmitted using a given wavelength from a node in the network will be communicated to all other nodes in the network.

Physical Topology: Provides the physical connections between the nodes in a network.

Regular Graph: A graph where each node has same node degrees.

Routing and Wavelength Assignment (RWA): Given a network topology and a set of lightpaths (to be determined), the problem of routing the lightpaths in the network and

assigning wavelengths to these lightpaths is referred as the routing and wavelength assignment (RWA).

Single-Hop Network: A network in which a packet travels from its source to its destination directly (in one hop). The packet does not encounter an electro-optic conversion before reaching its final destination.

Static Interconnection Network: All connections among the processors are fixed meaning that the processors are wired directly.

Wavelength Division Multiplexing (WDM) Network: It is a promising approach used in optical fiber where the optical transmission spectrum is divided into a number of non overlapping wavelength (or frequency) bands, with each wavelength supporting a single communication channel operating at peak electronic speed.

Wavelength Routed Network: A wavelength routed WDM network is a network where each end-node (the source or destination of data) is connected to a router and each router is connected to other routers. The advantage of such network is that the data is not broadcast to all the end-nodes. The settings of the routers determine which end-nodes will be connected by a lightpath.

Bibliography

[Ak89] S. G. Akl, The Design and Analysis of Parallel Algorithms, The Prentice-Hall Inc., New Jersey, 1989.

[BBRM97] M.S Borella, J.P. Jue, D. Banerjee, B. Ramamurthy and B. Mukherjee, "Optical components for WDM lightwave networks", Proceedings of the IEEE, Vol. 85, Issue 8, pp. 1274 -1307 , 1997.

[Be73] C. Berge, Graphs and Hypergraphs, American Elsevier publishing company, Inc., 1973.

[ChKr93]R.D. Chamberlain and R.R Krchnavek, "Architectures for optically interconnected multicomputers", IEEE Global Telecommunication Conference, GLOBECOM '93, Vol. 2, pp. 1181-1186, 1993.

[CrBo98] O. Crochat, and Jean-Y.L. Boudec. "Design Protection for WDM Optical Networks", IEEE J. Select Areas Commun. Vol. 16, No. 7, pp. 1158-1165, 1998.

[DDHH+99] B. T. Doshi, S. Dravida, P. Harshavardhana, O. Hauser, and Y. Wang, "Optical network design and restoration", Bell Labs tech. J., pp. 58 – 83, 1999.

[DDS99] D. Das, M. De and B. P. Sinha, "A new network topology with multiple meshes", IEEE Transactions on Computers, Vol. 48, No. 5, pp. 536-551, 1999.

[DGS97] M. De, D. Das, M. Ghosh and B. P. Sinha, "An efficient sorting algorithm on the Multi-Mesh network", IEEE Transactions on Computers, Vol. 46, No. 10, pp. 1132-1137, 1997.

[EBC98] G. Ellinas, K.Bala and G. K. Chang, "Scalibility of a novel wavelength assignment algorithm for WDM shared protection rings", Proc. IEEE/OSA'98 Optical Fiber Communication Conference, pp. 363-364, 1998.

[Ge98] O. Gerstel, "Opportunities for optical protection and restoration", in Optical Fiber Communication Conf., Vol. 2, San Jose, CA, pp. 269 – 270, 1998.

[GeRa00] O. Gerstel and R. Ramaswami, "Optical Layer Survivability: A Services Perspective", IEEE Commun. Mag., pp. 104-113, 2000.

[HKMF+95] W. Hendrick, O Kibar, P. Marchand, C. Fan, D. V. Blerkom, F. McCormick, I. Cokgor, M. Hansen and S. Esener, "Modeling and optimization of the optical transpose interconnection system", in Optoelectronic Technology Center, Program Review, Cornell University, 1995.

[HwBr83] K. Hwang and F. A. Briggs, Computer Architecture and Parallel Processing. New York: McGraw-Hill, 1983.

[Le92] T. Leighton, Introduction to Parallel Algorithm and Architectures: Arrays · Trees · Hypercubes, Morgan Kaufmann Publishers, San Mateo, California, 1992.

[LiFi01] L. L. Ling and A.J.C. Filho,"D-ARM: A new proposal for Multi-Dimensional interconnection networks", SIGCOMM Computer Communication Review, Vol. 31, No. 1, 2001.

[LoSu94a] Louri and H. Sung, "An Optical Multi-Mesh Hypercube: a scalable optical interconnection network for massively parallel computing", Journal of Lightwave Technology, Vol.12, Iss. 4, pp. 704 -716, 1994.

[LoSu94b] Louri and H. Sung, "3D Optical Interconnects for high-speed interchip and interboard communications", Computer, Vol. 27, pp. 27–37, 1994.

[MMHE93] G. C. Marsden, P. J. Marchand, P. Harvey and S. C. Esener, "Optical transpose interconnection system architectures", Optical Letters, Vol. 18, No. 13, pp. 1083-1085, 1993.

[MPPM02] G. Maier, A. Pattavina, S. D. Patre, and M. Martinelli, "Optical Network Survivability: Protection Techniques in the WDM Layer", Photonic Network Communication, 4:3/4, 251-269, 2002.

[Mu97] B. Mukherjee," Optical Communication Networks", McGraw-Hill, Newyork, 1997.

[MuSe03] S. Murthy and A. Sen, "A Peer-to-Peer Network Based on Multi-Mesh Architecture", GLOBOCOM, 2003

[Os00] A. Osterloh, "Sorting on the OTIS-Mesh", Proc. 14th International Parallel and Distributed Processing Symposium (IPDPS 2000), pp. 269-274, 2000.

[Ra92] S. Rajasekaran, "Randomized algorithms for packet routing on the Mesh", Advances in Parallel Algorithms, pp. 227-301, 1992.

[RaMu99a] S. Ramamurthy and B. Mukherjee, "Survivable WDM Mesh Network, Part I-Protection", Proc. of IEEE INFOCOM '99, pp.744-751, 1999.

[RaMu99b] S. Ramaswami and B. Mukherjee, "Survivable WDM mesh networks Part II – Restoration", ICC'99. 1999 IEEE International Conference on Communications, Vol. 3. pp. 2023-2030, 1999.

[RaSa98] S. Rajasekaran and S. Sahni, "Randomized routing, selection and sorting on the OTIS-Mesh optoelectronic computer", IEEE Transactions on Parallel and Distributed Systems, Vol. 9, No. 9, pp. 833-840, 1998.

[SaWa97] S. Sahni and C.-F. Wang, "BPC permutations on the OTIS-Mesh optoelectronic computer", Proc. of the 4th International Conference on Massively Parallel Processing using Optical Interconnections (MPPOI '97), Montreal, Canada, pp. 130-135, 1997.

[SBS01] A. Sen, S. Bandyopadhyay and B. P. Sinha, "A new architecture and a new metric for lightwave networks", IEEE Journal of Lightwave Technology, Vol. 19, No. 7, pp. 913-925, 2001.

[ScSe89] I. D. Scherson and S. Sen, "Parallel sorting in two-dimensional VLSI models of computation", IEEE Transactions on Computers, Vol. 38, No. 2, pp. 238-249, 1989.

[SFK97] D. Sima, T. Fountain, P. Kacsuk, "Advanced Computer Architectures", Addison Wesley, 1997.

[So03] V.Soini, "Introduction to Optical Networking", 2003.

[SRM02] L. Sahasrabudhe, S. Ramamurthy and B. Mukherjee, "Fault management in IP-over-WDM networks: WDM protection versus IP restoration", IEEE J. Select. Areas Commun., Vol. 20, No. 1, pp. 21 – 33, 2002.

[St83] Q. F. Stout, "Mesh connected computers with broadcasting," IEEE Trans. Computers, vol. 32, pp. 826-830, 1983.

[StBa99] T. E. Stern, K. Bala, Multiwavelength Optical Network, United States of America, Addison Wesley Longman, Inc. 1999.

[StCo91] H.S.Stone andJ.Cocke,"Computer Architecture in the1990s," Computer Vol. 24, No. 9, pp30-38, 1991.

[Sz95] T.Szymanski,"Hypermeshes": Optical Interconnection Networks for Parallel Computing, 1995.

[To94] M. Tompa, Lecture notes on message routing in parallel machines, Technical Report # 94-06-05, Department of Computer Science and Engineering, University of Washington, 1994.

[Ul84] J. D. Ullman, Computational Aspects of VLSI, computer Science Press, 1984.

[WaSa00] C.-F. Wang and S. Sahni, "Image processing on the OTIS-Mesh optoelectronic computer", IEEE Transactions on Parallel and Distributed Systems, Vol. 11, No. 2, pp. 97-109, 2000.

[WaSa01] C.-F. Wang and S. Sahni, "Matrix multiplication on the OTIS-Mesh optoelectronic computer", IEEE Transactions on Computers, Vol. 50, No. 7, pp. 635-646, 2001.

[WaSa02] C.-F. Wang and S. Sahni, "Computational Geometry on the OTIS-Mesh opto-electronic computer", Proc. International Conference on Parallel Processing, Vol. 18, No. 21, pp.501-507, 2002.

[WaSa98] C.-F. Wang and S. Sahni, "Basic operations on the OTIS-Mesh optoelectronic computer", IEEE Transactions on Parallel and Distributed Systems, Vol. 9, No. 12, pp. 1226-1236, 1998.

[ZMPE00] F. Zane, P. Marchand, R. Paturi and S. Esener, "Scalable network architectures using the optical transpose interconnection system (OTIS)", Journal of Parallel and Distributed Computing, Vol. 60, No. 5, pp. 521-538, 2000.

[Zo96] P. Zoeteweij, "Multicomputer Routing Algorithms: A simulation study", M.Sc. Thesis, 1996.

VITA AUCTORIS

Nahid Afroz was born in Barisal, Bangladesh. She obtained a B.Sc. with honors, in Computer Science in 1999 from the University of Dhaka, Bangladesh. She is pursuing her graduate studies at the University of Windsor in Computer Science and is currently a candidate for the master's degree and hoping completion in June, 2004. Her thesis advisor is Dr. Subir Bandyopadhyay.